Secure Bank Application

Presented to:

Mr. Akram Taha Zeyad

Presentation by:

Mustafa Mohammed Abdulhadi

Yousif Najim Abdkhadhim

Yousif Mahmoud Shihab

The Vision: Fortifying Digital Finance

The Problem: Digital Banking Vulnerabilities

In an increasingly digital world, traditional banking faces significant security challenges. Users demand convenience, but this often comes at the cost of robust protection against cyber threats. The goal is to address common vulnerabilities inherent in digital financial transactions, ensuring user data integrity and privacy.

Our Goal: A Secure, Seamless Banking Experience

This project aims to develop a secure digital banking application that prioritizes user safety without compromising on functionality. Key objectives include implementing ironclad login mechanisms, securely managing user balances, and ensuring the integrity of every transaction. Our focus is on building trust through transparent and strong security measures.

System Architecture: A Unified Approach

Frontend: Streamlit

Streamlit provides an intuitive and interactive user interface, allowing for rapid development and deployment of web applications. Its simplicity enables us to focus on the user experience while ensuring a clean and responsive design for all banking interactions.

Backend: Supabase

Supabase serves as our robust backend, offering a powerful PostgreSQL database, authentication services, and real-time capabilities. Its comprehensive suite of tools simplifies development and provides a scalable, secure foundation for our application's data storage and management.

Security Layer

The security layer is the bedrock of our application, integrating advanced cryptographic techniques like AES for data encryption and JWT for session management. This multi-layered approach safeguards sensitive information and ensures authenticated, secure communication between the frontend and backend.

Data Foundation: Securely Stored Information

The application's integrity relies heavily on how its data is structured and protected. Our Supabase backend is meticulously designed to handle sensitive financial information with the highest security standards.

User Credentials

User authentication details, including usernames and securely hashed passwords, are stored in dedicated tables. Password hashing with Bcrypt ensures that even if the database were compromised, passwords remain unreadable.

Encrypted Balances

Account balances are critical financial data. These are not stored in plaintext but are robustly encrypted using AES. This ensures that a direct database query would only reveal ciphertext, preventing unauthorized access to financial figures.

Encrypted Transactions

Every transaction record, detailing transfers, deposits, and withdrawals, is also encrypted with AES. This provides an additional layer of security, making the history of financial movements opaque to any unauthorized entity.

All data is meticulously organized within Supabase tables, leveraging its built-in security features and row-level security policies to restrict data access based on user roles and permissions. This systematic approach forms a secure and reliable foundation for the entire banking application.

Fortifying the Core: Advanced Security Design

AES Encryption

Advanced Encryption Standard (AES) is employed for sensitive data like account balances and transaction details. This symmetric encryption algorithm is renowned for its strength and efficiency, offering robust protection against unauthorized data access.

Bcrypt Hashing

User passwords are never stored directly. Instead, they are hashed using Bcrypt, a cryptographic hash function designed to be computationally intensive. This makes brute-force attacks significantly more difficult, even if the hash is compromised.

Secure Key Storage

The encryption keys vital for AES operations are stored in highly secure, restricted environments, separate from the main database. This prevents a single point of failure and ensures that even if the database is breached, the keys remain protected.

Operational Blueprint: Logic and Flow

The system's logic orchestrates a seamless yet secure user experience, from initial access to active session management. Each component plays a crucial role in maintaining security and functionality.

01

Authentication Flow

When a user attempts to log in, the provided password is hashed and compared against the stored Bcrypt hash. Upon successful verification, a unique JSON Web Token (JWT) is issued, granting access to the application's features.

02

Encryption & Decryption Logic

Sensitive data—such as account balances and transaction specifics—is automatically encrypted using AES before being stored in Supabase. When this data is retrieved for display, it is securely decrypted on the server-side, ensuring that plaintext never leaves a secure environment.

03

JWT Session Handling

JWTs are used for managing user sessions. These tokens contain digitally signed information, allowing the backend to verify the user's identity and permissions without needing to re-authenticate on every request. JWTs have a limited lifespan and are securely transmitted, enhancing session security.

User Journey: From Registration to Transaction

Registration

Users create an account by providing necessary details. Passwords are immediately hashed using Bcrypt before storage.

Login

Authenticated users receive a JWT, enabling secure access to their banking dashboard.

Dashboard

Users view their encrypted balance, transaction history, and initiate new transfers.

Transfer

Users can initiate money transfers to other accounts, subject to balance validation.

Balance Update

Balances are updated and re-encrypted after successful transactions, ensuring real-time security.

Logout

User sessions are terminated, invalidating the JWT and ensuring account security.

Precision in Finance: Transaction Handling

Robust Balance Validation

Before any funds are transferred, the system performs a real-time validation of the sender's available balance. This critical step prevents overdrafts and ensures that transactions can only proceed if sufficient funds are present.

- Real-time balance check
- Pre-transaction fund verification
- Prevents insufficient fund errors

Encrypted Record Storage & Error Prevention

All transaction details—sender, receiver, amount, and timestamp—are encrypted using AES before being stored in the Supabase database. This guarantees the confidentiality and integrity of each financial movement. Furthermore, the system incorporates comprehensive error handling to address potential issues during the transaction process, ensuring data consistency and preventing financial discrepancies.

- AES encryption for all transaction data
- Immutable and auditable records
- Robust error handling mechanisms

Achievements: A Secure Digital Bank

Secure Authentication

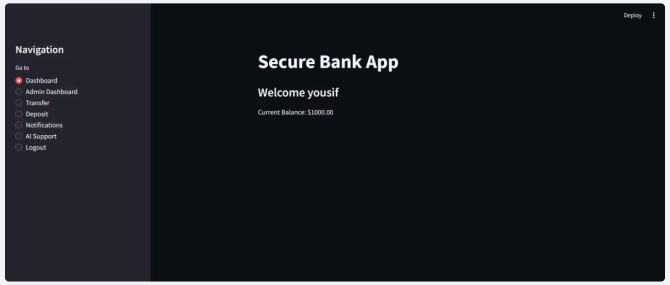
Implemented strong user authentication using Bcrypt for password hashing and JWT for session management, ensuring only authorized users can access accounts.

Protected Data Storage

Sensitive financial data, including balances and transaction details, are encrypted with AES, safeguarding information even in the event of a database breach.

Successful Transactions

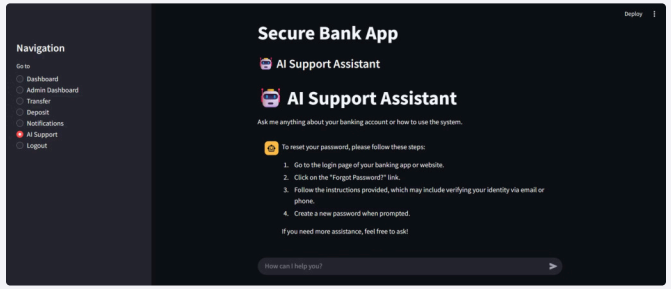
Developed and validated a robust transaction system with balance checks and encrypted record-keeping, ensuring reliable and secure fund transfers.



Looking Forward: Evolution of Secure Banking

Security-Focused Design

The foundation of this application is a security-first mindset, ensuring that every feature and interaction is built upon robust protective measures. This approach is paramount for financial applications where trust and data integrity are non-negotiable.



Scalability Potential

Built with Supabase and Streamlit, the application is designed for scalability, capable of handling a growing number of users and transactions without compromising performance or security. This modular architecture allows for easy expansion.

Future Enhancements

- Multi-Factor Authentication (MFA): Implement additional verification steps for enhanced security.
- Biometric Integration: Add fingerprint or facial recognition for quicker, more secure logins.
- Real-time Fraud Detection: Integrate AI-powered anomaly detection for suspicious activities.
- Advanced Reporting: Provide users with detailed insights into their spending habits and financial health.
- Open Banking APIs: Explore secure integration with other financial services.