Online Banking System

Semester Project

(BSCS)

By

S#	Name	Registration #/Roll #/Section	Mobile #	E-Mail
1.	ATEEQ UR REHMAN	Fa-2021/BSCS/197-E	+92 312 5053926	ateeq786316@gmail.com

Instructor:
Dr. Abdul Sattar ________________

Department of Computer Science
Lahore Garrison University
Lahore

Abstract:

The Online Banking System is a secure and user-friendly platform designed to streamline financial management. This online banking system lets users do important things like view balance, transfer money between accounts, deposit money to their accounts, and withdraw money out. It's built with the latest technology MERN (like React, Node.js, and MongoDB), which makes it work really well, handle lots of users, and look good on phones. The system is very secure with extra passwords, secret codes to protect your information, and follows strict safety rules. It's designed to be safe without being difficult to use. The system is carefully tested to make sure it's reliable and works smoothly, even when many people use it at the same time. This makes it a great solution for online banking that is both safe and easy to use.

Introduction:

The Online Banking System addresses the growing need for secure, efficient, and user-friendly digital financial platforms. With increasing reliance on online services, users demand seamless access to core banking features such as account management, fund transfers, deposit, withdraw and view balance, all while ensuring data security and compliance with financial regulations. This project aims to design and develop a robust platform that balances high-level security with optimal performance and usability, offering customers a reliable and intuitive solution for managing their financial activities across devices.

Literature Review

Online banking has made it much easier to manage our money. We can easily move money between accounts, pay bills, and keep track of our finances. However, many online banking systems have some problems. They may not be completely safe, they might not be able to handle a lot of users at once, and they can be difficult to use. For example, it can be hard to log in, the information might not be protected well enough, and the system can be slow when many people are using it at the same time. These issues can make it frustrating for customers.

The Online Banking System distinguishes itself by addressing these limitations with innovative features. It incorporates a secure and streamlined signup and sign-in process, multi-factor authentication for enhanced security, and real-time updates for transactions such as deposits, withdrawals, and transfers. The system's responsive design ensures compatibility across devices, while its efficient backend powered by modern technologies ensures scalability and fast performance. These innovations make the platform a reliable and user-centric solution for modern digital banking.

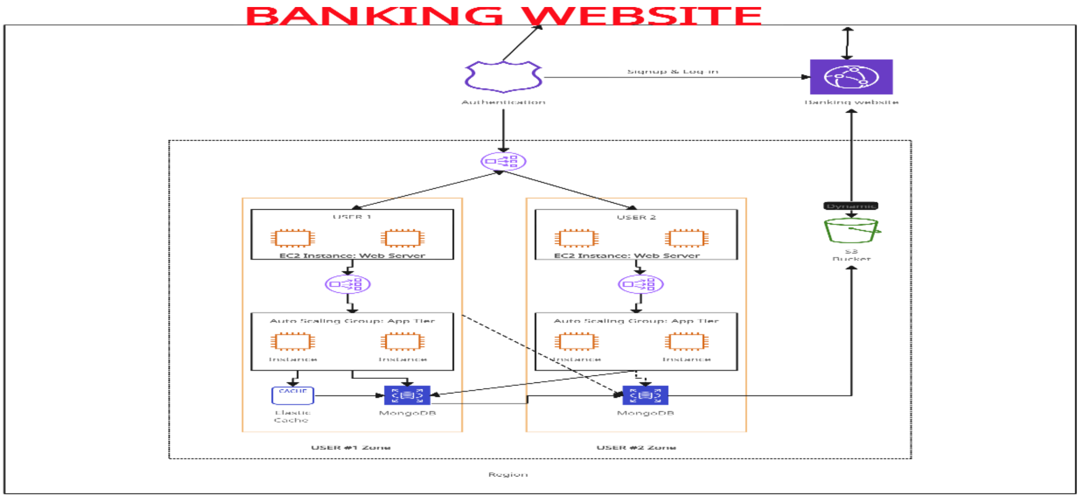
System Analysis

1. Signup and Sign-in: Users must be able to securely register for an account and log in using their credentials.
2. Account Management: Users can view their account balances, transaction history, and other relevant financial data.
3. Deposit: Users can deposit funds into their accounts, with real-time balance updates.
4. Withdraw: Users can withdraw funds from their accounts, and the system will reflect the updated balance immediately.

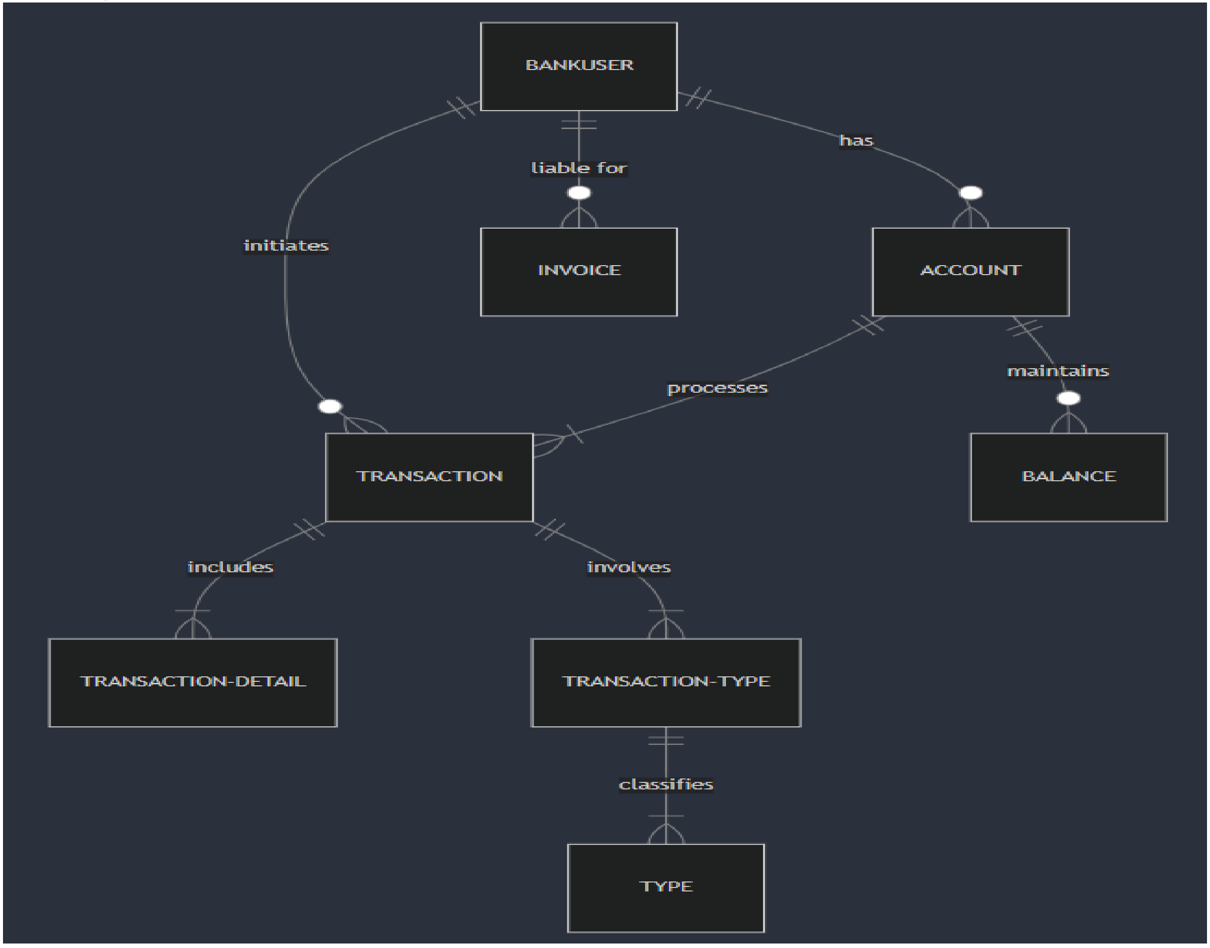
5. Fund Transfer: Users can transfer funds between accounts or to external accounts, with appropriate validation and confirmation steps.
6. Balance: Users can view a balance of his account.

System Design:

- Architecture Diagram (multi-tier architecture)



- ER Diagram for the database



Implementation:

Technologies Used:

Frontend: React.js, HTML5, Bootstrap JavaScript for building a dynamic, responsive UI.

Backend: Node.js, Express.js for server-side logic and API development.

Database: MongoDB for storing user data and transaction history.

Core Features Implemented:

Signup & Sign-in: Secure user registration and login.

Account Management: View balances, recent transactions, and account details.

Deposit & Withdraw: Users can deposit and withdraw funds, with real-time updates.

Fund Transfer: Secure internal and external transfers.

Mobile Responsiveness: Optimized design for all devices

Testing:

Testing included unit, integration, security, and performance assessments. Unit tests ensured individual components like login and fund transfers worked correctly. Integration tests confirmed seamless communication between the frontend and backend. Security tests identified vulnerabilities, using tool Postman. Test cases for login, fund transfers, deposits, and security checks all passed, demonstrating the application's reliability, security, and performance under load.

Challenges and Solutions:

MongoDB Connection Issue: Initially, the MongoDB connection failed due to using the wrong address (mongodb://localhost:27017/bankuser). It was resolved by using the correct connection string (mongodb://127.0.0.1:27017/bankuser), ensuring a stable connection to the database.

Port Mismatch: The project initially used port 3001 for the backend, but some configurations mistakenly referred to port 5000, causing errors. This was fixed by standardizing the port usage across the application to ensure consistent backend communication.

Conclusion:

The online banking project was successfully developed and implemented, providing users with a secure and user-friendly platform to manage their finances. Key features such as account creation, login, fund transfers, deposits, withdrawals, and balance checks were seamlessly integrated, with strong security measures in place, including encryption and authentication. Testing validated the platform's functionality and security, ensuring the system can handle high traffic while maintaining data integrity and regulatory compliance.

The project not only met the functional requirements but also delivered an optimized user experience, contributing to the institution's digital banking transformation.

For future enhancements, the platform could introduce two-step verification for added security, further improving user authentication. Additionally, features such as viewing transaction details and transaction history could be added to offer users greater transparency and control over their finances. Scalability options could include implementing cloud infrastructure to handle increased user traffic and utilizing machine learning for fraud detection and personalized banking experiences. These enhancements would not only improve security and usability but also ensure the platform remains robust and adaptable to future technological advancements.

References:

1. React Documentation. (n.d.). Retrieved from https://reactjs.org/docs/getting-started.html
2. Node.js Documentation. (n.d.). Retrieved from https://nodejs.org/en/docs/
3. MongoDB University. (n.d.). Learn MongoDB for free. Retrieved from https://university.mongodb.com
4. W3Schools. (n.d.). HTML, CSS, JavaScript Tutorials. Retrieved from https://www.w3schools.com
5. Mozilla Developer Network (MDN). (n.d.). Comprehensive Web Development Resources. Retrieved from https://developer.mozilla.org
6. Express.js Documentation. (n.d.). Retrieved from https://expressjs.com/
7. Postman. (n.d.). API Testing Tools. Retrieved from https://www.postman.com/
8. YouTube Tutorial on Secure Online Banking Website Development. (2020). Retrieved from https://www.youtube.com/watch?v=ZVyIIyZJutM
9. AWS Documentation. (n.d.). Retrieved from https://aws.amazon.com/documentation/
10. "Secure Coding Practices for Web Development." (2020). OWASP Foundation. Retrieved from https://owasp.org/www-project-top-ten/
11. Google Cloud Platform. (n.d.). Cloud Computing Services. Retrieved from https://cloud.google.com/docs