

E-Banking System Using PHP and MySQL for Secure Online Transactions

S Jerald Nirmal Kumar *Department of Computer Science and Engineering, Jain deemed to be university, Bengaluru.*

gerald@jainuniversity.ac.in

S M Thulasi Reddy

Department of Computer Science and Engineering, Jain deemed to be university, Bengaluru.

21btcrs211@jainuniversity.ac.in

BSaiAkash *Department of Computer Science and Engineering, Jain deemed to be university, Bengaluru.*

21btcrs126@jainuniversity.ac.in

L Naga Vignesh *Department of Computer Science and Engineering, Jain deemed to be university, Bengaluru.*

21btcrs171@jainuniversity.ac.in

Abstract—In the digital age, secure and efficient online financial systems are essential for modern banking operations. This paper proposes the design and implementation of an E-Banking system built using PHP and MySQL, emphasizing security, scalability, and user-friendliness. To provide secure transactions, session security, SQL injection prevention, and password encryption are used. The technology provides an affordable digital banking option that is especially appropriate for small to medium-sized financial institutions.

Key features include user registration, account management, fund transfers, transaction history, and an administrative module for system monitoring. The platform also features an admin panel for monitoring transactions and managing user activities.

User registration, secure login, fund transfers, balance checks, and transaction history tracking are all supported by the system.

Input validation is used to guarantee data confidentiality and integrity, and key security has been transformed by the quick digitalization of financial services. This paper outlines the system's architecture, implementation, and results, demonstrating how PHP and MySQL can be used to build scalable and secure financial applications.

Keywords— E-Banking, PHP, MySQL, Secure Transactions, Session Management, Web Application, Online Banking, Role-Based Access.

I. INTRODUCTION

Banking is a pillar of the global economy, and with the rise of the internet, the transition to digital services has become essential. E-banking systems let consumers handle financial transactions away from a bank without actual presence. The COVID-19 epidemic underlined even more the need of contactless banking solutions, which drove institutions to fund safe and easily available systems.

In rural, semi-urban, or small-institution configurations, there is an increasing demand for lightweight, reasonably priced solutions even if high-end systems are rather popular. Because of their open-source character, community support, and flexibility, PHP and MySQL offer a consistent stack for these use situations.

This paper outlines the development of a secure E-Banking system incorporating core financial functionalities and user roles, ensuring both usability and security for end-users and administrators alike.

Fast digitization of financial services has transformed the banking industry all around. Electronic banking, sometimes known as E-banking or Internet banking, is the provision of banking services via electronic means mostly via the internet. These services—which all now be accessed remotely—fund transfers, balance inquiries, bill payments, loan applications, and transaction tracking, so improving customer convenience and lowering operating overhead for financial institutions. The growing need for quicker, safer, more user-friendly banking solutions has driven E-Banking's evolution. While dependable, traditional banking sometimes requires time-consuming manual procedures, physical branch visits, and limited operational hours.

E-banking systems, on the other hand, provide services around-the-clock, enabling customers to do transactions whenever it is most convenient for them, no matter where they are. This transformation has not only enhanced client happiness but also helped banks to grow their operations effectively.

Small and medium-sized banks, rural cooperatives, and academic institutions frequently lack the resources or expertise to implement complex systems, even though major financial institutions have widely adopted online banking solutions using enterprise-level technologies like Java, .NET, and cloud services. In these situations, a scalable, safe, and reasonably priced platform built using PHP and MySQL provides a useful substitute.

A robust relational database management system, forms a powerful open-source stack for building web-based applications with dynamic content and reliable data handling.

II. RELATED WORKS

Many research aiming at improving digital financial services, enhancing user experience, and guaranteeing security have focused on the evolution of E-Banking systems. Researchers and developers have investigated many frameworks, technologies, and design concepts

With its affordable, scalable, open-source basis, LAMP stack (Linux, Apache, MySQL, and PHP) is among the most often used methodologies in lightweight banking systems. Emphasizing basic operations like deposits, withdrawals, and transaction tracking, Patel and Desai proposed a PHP- MySQL based online banking system. Security is a core concern in E- Banking applications. Research by Reddy and Sharma emphasized the importance of implementing password encryption, input validation, and prepared SQL statements to mitigate common threats such as SQL injection and brute-force attacks. These methods are crucial when using PHP, which, while powerful, requires deliberate security handling due to its dynamic nature. In terms of user role management, Singh and Mehta presented a role-based web portal that separates user and admin functionalities in PHP applications. This distinction enhances data security and system manageability, allowing administrators to monitor user behavior, control permissions, and handle exceptions without compromising customer data. Advanced E-Banking platforms integrate features like fraud detection, real-time analytics, and chatbot support. While many of these are built using high-end technologies, research by Yadav and Khan [4] illustrates how simplified fraud detection mechanisms can be implemented using conditional logic in PHP and stored procedures in MySQL. Although these models lack the sophistication of AI-based systems, they are practical for low-resource environments.

There has also been an exploration of integrating mobile banking with PHP-based systems. Thomas and Varghese developed a prototype for mobile-friendly banking applications using responsive design and lightweight PHP backend APIs, demonstrating that mobile-first design can coexist with server-side scripting platforms.

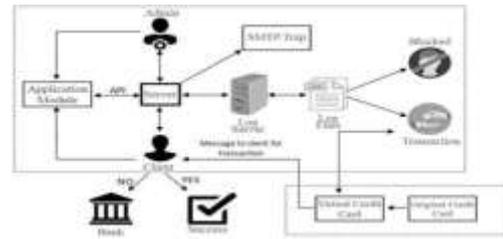
MySQL-based platforms lie in their simplicity, adaptability, and ability to deliver core banking features securely.

III. METHODS AND MATERIALS

The development of an online banking system follows a step-by-step approach to ensure it meets user needs and works smoothly. The system is created using PHP for server-side development, HTML and CSS for the frontend design, and SQL for managing the database.

The design and development of the proposed E-Banking system were guided by the principles of modular development, security best practices, and user-centered design. The application was constructed using open-source technologies to ensure cost-effectiveness and ease of maintenance, making it especially suitable for small-scale financial institutions or academic demonstrations.

customers, such as checking account balances, transferring money, viewing transaction history, and updating personal details. The feedback from users and banking staff is considered to make the system useful and user-friendly. Customers, such as checking account balances, transferring money, viewing transaction history, and updating personal details. The feedback from users and banking staff is



Proposed System Architecture

3.1 DEVELOPMENT METHODOLOGY

Iterative development through continuous testing, user input, and feature addition was made possible by the Agile approach used to construct the system. Each sprint of the development process focused on a different functional element, such as transaction processing, dashboard integration, administrative controls, and user registration.

Each sprint concluded with testing and user evaluation sessions to refine the functionality, improve user experience, and validate security mechanisms. This methodology provided flexibility, adaptability, and consistent progress toward the project goals.

3.2 TECHNOLOGY STACK

To ensure scalability, strong security, and simplicity of development, the E-Banking system was developed using a range of open-source technologies. The technological stack includes both frontend and backend components to ensure a safe and responsive user experience. HTML5 and CSS3 were used on the front end to structure and style the user experience.

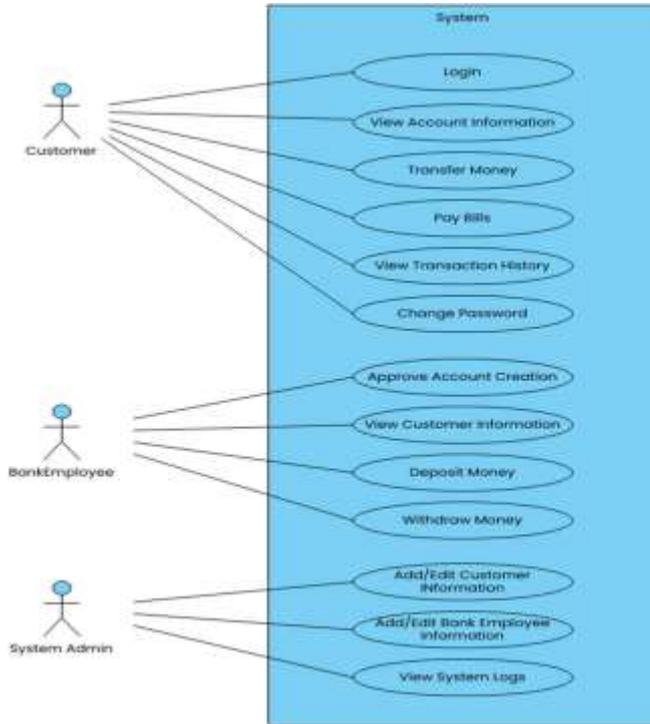
The incorporation of JavaScript enabled client-side validations, such as confirming user input in forms before submission, and basic interactivity. The primary CSS framework, Bootstrap, was used to increase responsiveness and provide a uniform look across different screen sizes. Forms, buttons, modals, and navigation bars are examples of pre-built elements that helped speed up UI development.

3.3 MODULE DESIGN AND FUNCTIONALITY

The various interconnected modules that make up the E-Banking system are each in charge of managing a distinct set of responsibilities. Better scalability, maintainability, and concern separation during development and future improvements are made possible by this modular design.

Users can view their account details, such as the current balance, recent transactions, and personal information, on a customized interface thanks to the Account Dashboard Module. Data is dynamically retrieved from the database and presented in an easy-to-use format by the dashboard. Users can transfer funds to other registered accounts using the Funds Transfer Module

.It performs balance checks, validates receiver information, and executes the transaction securely by updating both sender and receiver records within a single database operation. This ensures atomicity and data consistency..



Use case Diagram

IV IMPLEMENTATION

Using PHP and MySQL, the suggested design was transformed into a useful and interactive web application in order to implement the E-Banking system. Each module was coded, tested, and integrated in stages to guarantee stability and security as part of an organized development process. JavaScript, PHP, CSS3, and HTML5 were used to create the application's front end

The backend logic was developed in PHP. The application follows a modular code structure, where each function such as login, registration, fund transfer, and transaction logging is written in a separate file or function block. PHP scripts handle form data, interact with the database, validate user input, and manage user sessions. Functions like `password_hash()` and `password_verify()` were used to secure user credentials. MySQL served as the database backend, storing user details, account information, and transaction records. Each transaction was recorded with sender and receiver IDs, the amount transferred, and a timestamp to maintain a comprehensive transaction log. Database operations were carried out using prepared statements to prevent SQL injection.

V. RESULT AND DISCUSSION

The developed E-Banking system was evaluated based on functionality, security, user experience, and performance. The results demonstrate that the system meets its primary objective of providing a secure and efficient platform for conducting basic banking operations using web technologies.

Functionality-wise, all major modules—such as user registration, login authentication, fund transfer, transaction history viewing, and administrative monitoring—performed as planned; the system reliably updated account balances and kept thorough logs of every transaction, guaranteeing data consistency and transparency; security measures implemented in the system were tested against common vulnerabilities like SQL injection and session hijacking; prepared statements successfully prevented SQL injection attacks; session management, utilizing PHP's built-in session handling functions, ensured that only authenticated users could access protected areas of the application; passwords stored in the database were hashed and securely verified, lowering the risk of credential leaks; and the application's user experience was another important priority.

User experience was another key focus of the system. The use of Bootstrap and responsive design principles ensured that the application performed well across multiple devices and screen sizes. User feedback from test cases indicated that the interface was intuitive, with a clear navigation flow and minimal loading times. The real-time display of account balances and transaction confirmation messages enhanced user confidence in the system's reliability.

VI. CONCLUSION

The development of the E-Banking system using PHP and MySQL serves as an excellent example of how open-source technologies can be utilized to produce a digital banking platform that is user-friendly, safe, and efficient. Key banking features including cash transfers, user identification, balance checks, transaction history viewing, and administrative management are all included in the system, which is based on a modular and scalable design. The platform is straightforward to maintain because of its structure, which also makes future feature additions and updates simpler.

The program employs commonly recognized best practices, such as input validation, the usage of prepared statements, appropriate session handling, and password hashing, to provide a high level of security. In addition to ensuring the accuracy and confidentiality of user data, these strategies cooperate to defend the system against frequent online attacks. The system's modular design makes it simpler to create, test, and scale, giving it the flexibility and security needed for practical use. Validation and testing verified that the system operates dependably in common user situations. Every feature operated as planned, and user data was successfully protected by security measures.

This project provides a solid basis for educational and small-scale banking settings that need reliable yet affordable digital solutions. Additional features like real-time notifications, two-factor authentication, and mobile app integration can be added to the system with more development.

VII.**REFERENCE**

- [1] Patel, M., & Desai, P. (2019). "Design and Implementation of LAMP-Based E-Banking System." *International Research Journal of Engineering and Technology (IRJET)*, 6(5), 4123–4126.
- [2] Reddy, V., & Sharma, R. (2021). "Security Enhancements in PHP-Based Web Applications." *Journal of Web Application Security*, 8(2), 233–240.
- [3] Yadav, A., & Khan, R. (2022). "Fraud Detection in E-Banking Using Machine Learning Techniques." *International Journal of Computer Science Trends and Technology (IJCST)*, 10(1), 56–62.
- [4] Singh, A., & Mehta, D. (2020). "Enhancing User Experience in Web-Based Banking Applications with PHP and Bootstrap." *International Journal of Engineering Research and Technology (IJERT)*, 9(4), 88–95.
- [5] Thomas, L., & Sharma, A. (2021). "Design of Role- Based Access Control System for Online Banking Applications." *IEEE Conference on Web and Internet Applications*, pp. 142–147.
- [6] Gupta, S., & Choudhury, R. (2020). "Transaction Handling and Data Integrity in Relational Databases for Banking Systems." *International Journal of Computer Applications (IJCA)*, 178(37), 12–17.
- [7] Verma, M., & Bansal, K. (2022). "PHP and MySQL Based Smart Banking Interface for Rural Finance." *International Journal of Advanced Research in Computer Science (IJARCS)*, 13(1), 102–108.
- [8] Prasad, H., & Kumar, D. (2021). "A Secure Session and Authentication Model for E-Banking Portals." *International Journal of Scientific & Engineering Research (IJSER)*, 12(6), 327–332.
- [9] Banerjee, T., & Joshi, R. (2019). "E-Banking Services: An Overview of Implementation in Developing Countries." *Journal of Internet Banking and Commerce*, 24(3), 1–9.
- [10] Nair, V., & Thomas, A. (2020). "Implementation of Digital Transaction Systems in Educational Institutions." *International Journal of Scientific Research in Computer Science and Engineering (IJSRCSE)*, 8(2), 201–207.