International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1152-1157
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0164

KYC Verification Service Using MERN Stack

Mr. Nagesh B. Mapari [1], Pragati S. Dhule[2], Rajendra G. Ingle[3], Punam R. Sardar[4], Saurabh R. Wankhade[5]
[1]Associate professor, Dept. of IT, Anuradha Engineering College, Chikhli, Maharashtra, India.
[2,3,4,5]UG Scholar, Dept. of IT, Anuradha Engineering College, Chikhli, Maharashtra, India.
Emails: *nagas7366@gmail.com[1], pragatidhule104@gmail.com[2], rajendraingle86@gmail.com[3], punamsardar791@gmail.com[4], wankhadesaurabh2003@gmail.com[5]*

Abstract
Know Your Customer (KYC) verification is a crucial process for businesses to validate user identities and assess potential risks. Traditional manual KYC methods, however, are often insecure, time-consuming, and costly. To address these challenges, this project proposes a KYC verification system using the MERN (MongoDB, Express, React, Node.js) stack. MongoDB are used to protect the data. The system ensures security, efficiency to protect sensitive user data. Built with the MERN stack, the system provides a scalable and efficient KYC verification process. Furthermore, it offers an intuitive interface for users to upload and manage their KYC documents, enhancing overall user experience.
Keywords: *User Authentication, KYC, MERN Technology, client websites and app.*

1. Introduction
Know Your Customer (KYC) verification is integrated into the MERN Stack application to validate user identities and assess potential risks of illegal intentions. Using MongoDB for data storage, Express.js for backend functionality, React.js for the user interface, and Node.js as the runtime environment, the application ensures a secure and efficient KYC verification process [1]. The existing systems of data storage have significant security concerns. To address these concerns, this project proposes a secure KYC verification system using the MERN (MongoDB, Express.js, React.js, and Node.js) Stack, MongoDB server are used to secure data storage [2]. The system also employs Axios for secure HTTP requests, JSON Web Tokens (JWT) for authentication and authorization, Bcrypt for password hashing, Multer for secure file uploads, and NodeMailer for secure email communication. The existing manual KYC process is time-consuming, redundant, costly, and less secure. For instance, in banking systems, customers with multiple accounts created at different time intervals face difficulties in updating their KYC information. The proposed system seeks to solve these problems by providing clients with control over their information, safety, efficiency gains, cost advantages, enhanced customer experience, and increased transparency throughout the onboarding process. The proposed system can be deployed in various organizations that require KYC verification of customers. The system's flexibility and scalability make it an attractive solution for businesses looking to improve their KYC processes. In addition to its benefits, the proposed system also has several features, including secure data storage, real-time updates, and scalability. The system's ability to handle large volumes of data and provide real-time updates makes it an ideal solution for organizations requiring KYC verification. The proposed system has several benefits, including enhanced security measures to protect sensitive KYC data, improved efficiency and scalability for KYC verification, reduced risk of data breaches and unauthorized access, and compliance with regulatory requirements for KYC verification [3]. the KYC Verification Services system built on the MERN stack can still ensure security, data integrity, and efficiency through alternative approaches. Instead of a decentralized ledger, MongoDB can be used with encryption to securely store KYC data while maintaining data integrity using Mongoose schema validation. Additionally, immutable audit logs can be implemented to track changes and ensure transparency. To achieve distributed and redundant data management, MongoDB replica sets can be

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1152-1157
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0164

utilized for data redundancy, while cloud-based backups ensure data availability. For authentication and access control, JSON Web Tokens (JWT) can be implemented for secure user authentication [4]. In place of smart contracts, server-side business logic in Node.js can be designed to automate KYC verification processes, enforcing validation rules and securely storing verification statuses in MongoDB. To prevent data tampering, cryptographic hashing techniques such as bcrypt can be applied to sensitive data, ensuring its integrity Transaction history can be stored as immutable logs to maintain a record of KYC verification activities. Through these measures, the MERN stack-based KYC Verification Services system can provide a secure and efficient alternative to blockchain-based implementations while maintaining data reliability and trustworthiness [5], [6],[7].

2. KYC Challenges

MERN stack-based KYC Verification system, security threats such as malware attacks, unauthorized data access, and cyber intrusions must be addressed effectively. One major challenge is protecting sensitive user data, as attackers may attempt to scan the KYC server for confidential files, manipulate stored data, or plant malicious code to gain real-time access. Cyber threats can modify file attributes, transfer KYC documents to unauthorized locations, execute harmful commands, or even delete all stored KYC information. To enhance security, multiple layers of protection can be implemented. The database is managed using MongoDB Atlas to ensure high availability and data security, while JWT authentication ensures secure user sessions. These security measures collectively enhance the integrity and confidentiality of KYC data in a MERN stack-based system, reducing the risk of cyberattacks and unauthorized access. [8].

3. Related Works

At present, electronic Know Your Customer (e-KYC) processes have become increasingly complex. The process involves secure credential registration, KYC document management, and secure verification processes between clients and financial institutions. Recent research works related to e-KYC focus on devising frameworks for secure user identity management and credentials verification. To address these challenges, this project proposes a secure e-KYC system using the MERN (MongoDB, Express.js, React.js, and Node.js) Stack. The system utilizes MongoDB for secure data storage, Express.js for backend functionality, React.js for the user interface, and Node.js as the runtime environment. The proposed system employs Axios for secure HTTP requests, JSON Web Tokens (JWT) for authentication and authorization, Bcrypt for password hashing, Multer for secure file uploads, and NodeMailer for secure email communication. These tools provide an additional layer of security, making it increasingly difficult for unauthorized access. The proposed system has several benefits, including enhanced security measures to protect sensitive KYC data, improved efficiency and scalability for KYC verification, reduced risk of data breaches and unauthorized access, and compliance with regulatory requirements for KYC verification [9],[10],[11].

3.1.MERN Stack Orchestration and Experimentation Framework

A Case Study of KYC The MERN Stack test bed is a new framework for studying the performance and evaluation of MERN-based applications in a real-world environment. It can be used to test with all available options to vary several parameters and offer perceptions on bottlenecks and evaluates MERN-based systems. It is also assessed for the use case of MERN-based KYC Proof of Concept (PoC) application. The paper concludes by discussing the various parameters which are to be considered before building a MERN-based solution. One of the most important aspects is to choose the right combination of MongoDB, Express.js, React.js, and Node.js to ensure a scalable and efficient solution. Additionally, considerations such as security, data storage, and user interface design are crucial in developing a successful MERN-based application[12]

3.2.Enhancing Privacy

Using MERN Stack to Protect Personal Data: The data security for preserving private data about individuals using MERN Stack was discussed. We ensures the privacy by collecting only the data required by kyc, by encrypting sensitive information during storage. A new method using a combination

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1152-1157
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0164

of MongoDB for secure data a storage, Express.js for backend functionality, React.js for user interface, and Node.js as the runtime environment is proposed. The issues addressed are such that data ownership, auditability, transparency, and fine-grained access controls. The proposed solution uses a combination of JSON Web Tokens (JWT) for authentication and authorization, Bcrypt for password hashing, and Multer for secure file uploads. It is concluded that the proposed solution enhances the security and privacy of personal data by implementing secure data storage and access controls. However, further testing is required to analyze the strengths and weaknesses of the proposed solution [13]

4. System Overview

This section describes the system model of our proposed e-KYC system using the MERN Stack. The system model consists of the following entities: authority, clients, financial institutes, MongoDB, Express.js, React.js, and Node.js.Authority: The authority generates the public parameter and the master private key of the system. The authority keeps the master private key secret and publishes the public parameter for the subscribers. Clients are the customers of financial institutes who join the e-KYC system. Each customer has their own key pair used to encrypt and decrypt their credential data. MongoDB is used as a secure database to store encrypted documents of KYC bound to each user account. Express.js is used for backend functionality, React.js for the user interface, and Node.js as the runtime environment.The system uses JSON Web Tokens (JWT) for authentication and authorization, Bcrypt for password hashing, and Multer for secure file upload The proposed system has three main components: (1) Registration component is responsible for authenticating users and enrolling new users, (2) Profile component is responsible for controlling client profiles and keeping track of user credentials, and (3) Verification component is responsible for e-KYC verification.In the next section, we describe two core processes of our system: client registration and e-KYC document uploading, and e-KYC verification. We describe the details of each process through the components developed for automating core e-KYC processes. [11]

4.1.Client Registration

- Step 1: Registration with Username and Email, Client provides username and email address, System checks foravailability of username and validity of email address .
- Step 2: Verification with Mobile or Email ID, System sends a verification code to the client's mobile number or email address, Client enters the verification code to verify their mobile number or email address.
- Step 3: Enter OTP (One-Time Password) :System generates an OTP and sends it to the client's verified mobile number or email address , Client enters the OTP to proceed with the registration process.
- Step 4: Verification of Documents, Client uploads their identification documents (e.g. ID card, passport, driving license), System verifies the authenticity of the documents.
- Step 5: Enter OTP (Again): System generates another OTP and sends it to the client's verified mobile number or email address, Client enters the OTP to confirm the verification of their documents.
- Step 6: Upload Documents: Client uploads additional documents required for registration (e.g. proof of address, income proof)System stores the uploaded documents in the client's.
- Step 7: Registration Completion: System confirms the completion of the client registration process ,Client can now access their account and use the system's services (Figure 1)

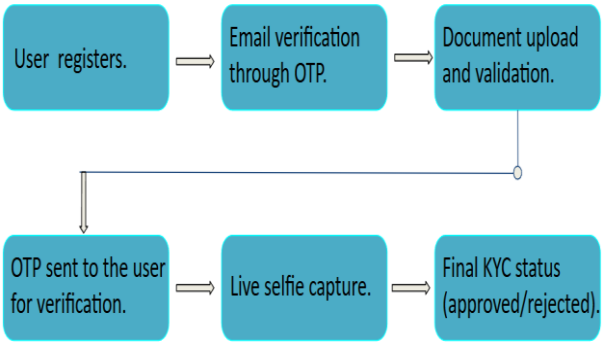

Figure 1 The Design Process

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1152-1157
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0164

5. Implementation

The proposed system is developed and tested using MongoDB, Express.js, React.js, and Node.js (MERN stack) to provide a secure and efficient Know Your Customer (KYC) verification process. This system ensures data security, encryption, and authentication while facilitating seamless user interactions. The backend is built using Node.js and Express.js, creating a robust API for handling KYC data, user authentication, and secure document storage. JWT (JSON Web Token) is implemented to ensure secure access control and session management. MongoDB are used for storing the document in the database, ensuring data privacy. To handle file uploads such as ID proofs and address verification documents, Multer is integrated. Secure password storage is achieved using bcrypt, which hashes and stores user passwords securely. Nodemailer is used to send verification emails containing secret keys to users in the form of OTP, while Axios facilitates seamless communication between the frontend and backend. The system's database is managed using MongoDB, which stores user details, encrypted KYC data, authentication tokens, and file metadata.The frontend, developed using React.js, offers an intuitive user interface for submitting KYC data, tracking verification status. The system supports secure login and signup with JWT authentication, KYC form submission for uploading identity documents, OTP generation and real-time API communication using Axios. The system follows a well-structured workflow, beginning with user registration and authentication using JWT and bcrypt. Once authenticated, users can submit their KYC data through a React-based form, with Multer handling document uploads. The submitted data is then encrypted using CryptoJS and securely stored in MongoDB. A verification email containing an OTP is sent to users via Nodemailer for added security. The admin verifies and approves the KYC data, after which users can access their KYC status through the system. This MERN-based KYC verification system provides a secure, efficient, and user-friendly approach to identity verification while ensuring the confidentiality and integrity of user data [14], [15], [16]. React.js makes it easy to create an interactive and user-friendly frontend. Node.js and Express.js are ideal for handling backend processes efficiently. MongoDB provides a flexible database for storing user data. The MERN stack is widely used in the industry and supports full-stack development with JavaScript, which improves efficiency. Security can be achieve JWT (JSON Web Tokens) for secure user authentication. Encrypting passwords using bcrypt. HTTPS for secure communication between the client and server. Ensuring that sensitive data like documents and selfies are stored securely in the database with access control. The live selfie capture feature works by: Allowing users to capture a selfie using their device camera. Comparing the selfie with the photo on their uploaded document using face-matching algorithms. If the two match, the KYC is approved; otherwise, it is rejected. KYC Verification Service is designed as a modular system. Businesses can integrate it into their apps or websites using APIs. We provide a detailed integration guide, and the APIs can be customized based on client needs. [12-13]

6. Future Work

The proposed KYC Verification System built using the MERN stack and AWS S3 for document storage effectively enhances identity verification and data security. However, there are several areas for future improvement and expansion. One key area for future work is the integration of AI-based facial recognition to enhance the selfie verification process. Currently, the system verifies the uploaded selfie manually or through basic matching algorithms, but incorporating AI/ML models can improve accuracy and reduce human intervention. Additionally, implementing liveness detection can prevent fraud attempts using static images. Another improvement is the implementation of decentralized identity verification using blockchain technology. By storing hashed KYC data on a blockchain, users can have more control over their personal information while insuring data intrigrity and security.This would eliminate the need for repeated KYC verification across multiple platforms. Moreover, enhanced document verification mechanisms can be introduced, such as OCR (Optical Character Recognition) for automated text extraction validation of identity documents. This

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1152-1157
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0164

would speed up the verification process and reduce manual errors. Security can also be further strengthened by incorporating multi-factor authentication (MFA), such as biometric authentication or one-time passwords (OTPs) for additional verification steps. This would help in mitigating unauthorized access and identity fraud. Lastly, to improve scalability and performance, the system can be migrated to a microservices architecture, allowing different components such as user authentication, document verification, and data storage to function independently. This would enhance system flexibility, maintenance, and support for a growing number of users. By incorporating these enhancements, the KYC Verification System can evolve into a more secure, efficient, and user-friendly identity verification platform, catering to the increasing demand for digital identity verification across industries. [14]

7. Observation and Result

With this paper we aimed and accomplished a solution that reduces the aggregated cost of the process of KYC in an ecosystem by means of Blockchain. We solved the first part of the problem by avoiding redundancy of tasks needed to be performed by the customer in case of multiple financial institutions. Moreover, we suggested following the verification process for one customer only once and maintaining it centrally. This not only helps the customer's in making their experience less cumbersome but also drastically reduces the cost of KYC process undertaken by the financial institutions in hiring third parties to carry out background checks, etc for their customers. Hence, the ultimate efficiency gain of our proposed solution was the dual benefit of reduced cost for the institutions and better experience for the customers. (Figure 2) These including above mention factor the result of the process may give 100% but excluding these factor and adding new factor may varying the result upto 5%. [15] enhance the selfie verification process. Currently, the system verifies the uploaded selfie manually or through basic matching algorithms, but incorporating AI/ML models can improve accuracy and reduce human intervention are used for storing the document in the database

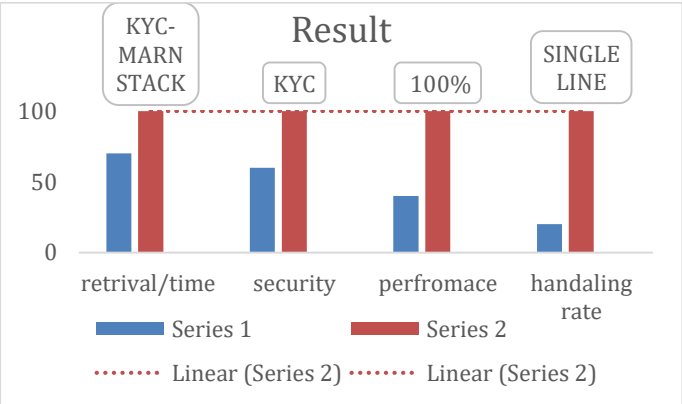

Figure 2 Result

Conclusion

The KYC Verification System developed using the MERN stack provides a secure, efficient, and scalable solution for identity verification. By integrating OTP-based authentication, secure document uploads or store using MongoDB, and user-friendly verification steps, the system ensures a seamless onboarding process for users while maintaining high security standards. The implementation of bcrypt for password hashing and JWT for authentication enhances security by protecting user credentials and session data. The use of MongoDB for document storage ensures scalability and efficient management of KYC documents, while the incorporation of Multer facilitates secure file handling. Additionally, the verification process, including document upload and live selfie capture, strengthens identity validation and prevents fraudulent access. The system not only improves the efficiency of KYC verification but also addresses security challenges associated with user authentication and document management. [16]

References

[1]. José Parra Moyano and Omri Ross, "KYC Optimization using Distributed Ledger technology", Springer -Business & Information systems Engineering, pp-411-423, vol.59,2018.

[2]. Cryptovest news desk, The Binance KYC leak reveals the need for government partnership, "http://cryptovest.com/news/the-binance-kyc-leak .

[3]. José Parra Moyano and Omri Ross, "KYC Optimization using Distributed Ledger

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1152-1157
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0164

technology", Springer -Business & Information systems Engineering, pp-411-423, vol.59,2018.

[4]. Mauro Isaja and John Soldatos, Distributed ledger technology for decentralization of manufacturing processes, IEEE Conference on Industrial Cyber-Physical Systems (ICPS) 2018 at, St. Petersburg, Russia, pp 696- 701, 2018.

[5]. Rui Yuan, Yu-Bin Xia, Hai-Bo Chen, Bin-Yu Zang, Jan Xie and ShadowEth: Private Smart Contract on Public Blockchain, Springer, Journal of Computer Science and Technology, Issue 3, pp 542–556. vol 33,2018.

[6]. XiaoqiLi, PengJiang, XiapuLuo, TingChen and QiaoyanWen, "A survey on the security of blockchain systems", Elsevier Future Generation Computer systems", Aug 2017

[7]. Sein Myung, Jong and Hyouk Lee, Ethereum smart contract-based automated power trading algorithm in a microgrid environment, Springer Journal of Super computing, PP 1-11,2018.

[8]. Fatimah Alkhudhayr, Shouq Alfarraj, Buthina Aljameeli, Salim Elkhdhiri, " Information security: A review of information security issues and Techniques," IEEE International conference on computer applications& Internet security,2019.

[9]. Y. Zhong, M. Zhou, J. Li, J. Chen, Y. Liu, Y. Zhao, and M. Hu, Distributed blockchain-based authentication and authorization protocol for smart grid, Wireless Commun. Mobile Comput., vol. 2021, pp. 115, Apr. 2021, doi: 10.1155/2021/5560621.

[10]. S. Y. Lim, P. T. Fotsing, A. Almasri, O. Musa, M. L. M. Kiah, T. F. Ang, and R. Ismail, Blockchain technology the identity management and authentication service disruptor: A survey, Int. J. Adv. Sci. Eng. Inf. Tech., vol. 8, pp. 17351745, Sep. 2018.

[11]. A. A. Mamun, A. Al Mamun, S. R. Hasan, S. R. Hasan, M. S. Bhuiyan, M. S. Bhuiyan, M. S. Kaiser, M. S. Kaiser, M.A.Yousuf, and M. A. Yousuf, Secure and transparent KYC for banking system using IPFS and blockchain technology, in Proc. IEEE Region Symp. (TENSYMP), Jun. 2020, pp. 348351.

[12]. PaulJ.Taylor,TooskaDargahi,AliDehghantanha,Reza,M.Parizi,Kim Kwang and RaymondChoo, A systematic literature review of blockchain cyber security. "digitalCommunication and networks", Elsevier Feb 2019.

[13]. WjatscheslavBaumung, and VladislavFomin, Framework for enabling order management process in a decentralized production network based on the blockchain-technology, Elsevier, Procedia CIRP, PP 456-460, vol 79.

[14]. J.Uthayakumar, T.Vengattaraman and P.Dhavachelvan, A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications, Elsevier, Journal of king saud university Computer and Information Sciences, pp 1-22, May 2018.

[15]. KenanKalajdzic, SamaherHusseinAli and AhmedPatel Rapid lossless compression of short text messages, Elsevier, Computer Standards & Interfaces, PP 53-59, vol 37, 2015

[16]. SuryaPrakashMishra,Col.GurmitSingh and RajeshPrasad, A review on compressed pattern matching, Elsevier, Perspectives in Science, PP 727-729, vol 8, 2016.