International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Basic Login System with Authentication using MERN Stack

Pranav Anand

Student, Computer Science and Engineering

pranavanand015@gmail.com

Mr. Nitin Pal

Department of Computer Science and Engineering

Abstract

In the modern digital era, secure authentication mechanisms play a vital role in protecting sensitive user data and ensuring authorized access to web applications. This research presents the design and development of a basic login system using the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. The system integrates JSON Web Tokens (JWT) and bcrypt hashing techniques to enhance security and ensure safe user authentication.

The study explores various aspects of authentication including user registration, login validation, password encryption, token-based authentication, and secure access to protected routes. Passwords are hashed using bcrypt before storing them in the database, which prevents exposure of sensitive data even if the database is compromised. JWT is used to maintain authentication in a stateless manner, improving scalability and performance of the system.

The performance of the system is evaluated based on response time, reliability, and resistance to common security attacks. The results demonstrate that JWT-based authentication provides an efficient and scalable solution for modern web applications. This research contributes to understanding the practical implementation of authentication systems and highlights the importance of secure coding practices in web development.

Keywords: Authentication, MERN Stack, JWT, Web Security, Login System

1. Introduction

With the rapid growth of web applications, secure user authentication has become a fundamental requirement. Authentication ensures that only authorized users can access system resources. Traditional authentication methods relied on sessions, but modern applications use token-based authentication for better scalability. This research focuses on implementing a secure login system using MERN stack technologies. With the rapid growth of web applications, secure user authentication has become a fundamental requirement. Authentication ensures that only authorized users can access system resources. Traditional authentication methods relied on sessions, but modern applications use token-based authentication for better scalability. This research focuses on implementing a secure login system using MERN stack technologies. With the rapid growth of web applications, secure user authentication has become a fundamental requirement. Authentication ensures that only authorized users can access system resources. Traditional authentication methods relied on sessions, but modern applications use token-based authentication for better scalability. This research focuses on implementing a secure login system using MERN stack technologies.

2. Literature Review

Authentication systems have evolved significantly over time. Session-based authentication stores user data on the server, which can lead to scalability issues. Token-based authentication using JWT eliminates this limitation by storing authentication data on the client side. Password security is achieved using hashing algorithms such as bcrypt. Authentication systems have evolved significantly over time. Session-based authentication stores user data on the server, which can lead to scalability issues. Token-based authentication using JWT eliminates this limitation by storing authentication data on the client side. Password security is achieved using hashing algorithms such as bcrypt. Authentication systems have evolved significantly over time. Session-based authentication stores user data on the server, which can lead to scalability issues. Token-based authentication using JWT eliminates this limitation by storing authentication data on the client side. Password security is achieved using hashing algorithms such as bcrypt.

3. Methodology

The system is designed using a structured approach that includes registration, login, token generation, and access control. User data is stored securely in MongoDB. The frontend is built using React.js, and the backend is implemented using Node.js and Express.js. Passwords are hashed and tokens are generated for secure communication. The system is

designed using a structured approach that includes registration, login, token generation, and access control. User data is stored securely in MongoDB. The frontend is built using React.js, and the backend is implemented using Node.js and Express.js. Passwords are hashed and tokens are generated for secure communication. The system is designed using a structured approach that includes registration, login, token generation, and access control. User data is stored securely in MongoDB. The frontend is built using React.js, and the backend is implemented using Node.js and Express.js. Passwords are hashed and tokens are generated for secure communication.

4. System Architecture

The architecture of the system consists of three main layers: frontend, backend, and database. The frontend handles user interaction, the backend processes requests, and the database stores user information securely. Communication between components is handled using REST APIs. The architecture of the system consists of three main layers: frontend, backend, and database. The frontend handles user interaction, the backend processes requests, and the database stores user information securely. Communication between components is handled using REST APIs. The architecture of the system consists of three main layers: frontend, backend, and database. The frontend handles user interaction, the backend processes requests, and the database stores user information securely. Communication between components is handled using REST APIs.

5. Model Implementation

Password hashing is implemented using bcrypt to ensure security. JWT tokens are generated during login and verified for each request. Middleware is used to protect routes and ensure only authenticated users can access them. Password hashing is implemented using bcrypt to ensure security. JWT tokens are generated during login and verified for each request. Middleware is used to protect routes and ensure only authenticated users can access them. Password hashing is implemented using bcrypt to ensure security. JWT tokens are generated during login and verified for each request. Middleware is used to protect routes and ensure only authenticated users can access them.

6. Evaluation Metrics

The system is evaluated using metrics such as response time, security strength, and reliability. Performance testing shows that the system handles multiple requests efficiently while maintaining security. The system is evaluated using metrics such as response time, security strength, and reliability. Performance testing shows that the system handles multiple requests efficiently while maintaining security. The system is evaluated using

metrics such as response time, security strength, and reliability. Performance testing shows that the system handles multiple requests efficiently while maintaining security.

7. Result and Discussion

The results show that the system successfully performs authentication and prevents unauthorized access. JWT-based authentication improves scalability and reduces server load. The discussion highlights advantages over traditional methods. The results show that the system successfully performs authentication and prevents unauthorized access. JWT-based authentication improves scalability and reduces server load. The discussion highlights advantages over traditional methods. The results show that the system successfully performs authentication and prevents unauthorized access. JWT-based authentication improves scalability and reduces server load. The discussion highlights advantages over traditional methods.

8. Conclusion

This research presents a secure and scalable login system using MERN stack technologies. JWT and bcrypt provide strong security features. Future improvements may include multi-factor authentication and OAuth integration. This research presents a secure and scalable login system using MERN stack technologies. JWT and bcrypt provide strong security features. Future improvements may include multi-factor authentication and OAuth integration. This research presents a secure and scalable login system using MERN stack technologies. JWT and bcrypt provide strong security features. Future improvements may include multi-factor authentication and OAuth integration.

Acknowledgement

The author expresses gratitude to the faculty and institution for their guidance and support in completing this research.

Author Statement

I declare that this research work titled “Basic Login System with Authentication” is my original work and has not been submitted elsewhere.

References

1. K. Jensen, “Introduction to Web Security,” 2020
2. JWT.io Documentation
3. NPM bcrypt Documentation
4. MongoDB Documentation
5. Node.js Documentation
6. Express.js Guide