Archives available at journals.mriindia.com

International Journal on Advanced Computer Theory and Engineering

ISSN: 2319-2526
Volume 14 Issue 01, 2025

Result Paper on Implementing Secure Authentication in Node.js for Web Applications

Prof. Y. L. Tonape, Kamble Namrata Surendra, Kale Pragati Ramchandra, Mane Divya Shrimant, Rajepandhare Sakshi Yogesh

Peer Review Information

Submission: 15 Feb 2025
Revision: 23 March 2025
Acceptance: 27 April 2025

Keywords

JWT
OAuth2
Passport.js
Bcrypt
Session Management

Abstract

Authentication and Authorization are the base of security for all the Technologies present in this world today. Starting from your smartphone where a user authenticates himself before he could access the data inside to Entering into the WhiteHouse, you must authenticate yourself, and based on that you are authorized.It could be dangerous to the users of the hacked website because their sensitive information like a credit card, bank account details, etc. could be sold in the black market of the “dark web”. The aim behind developing an authentication system is to keep users’ sensitive information safe so that hackers cannot steal and sell the information on the dark web’s back market.

Introduction

This project aims to address the challenges of implementing secure and scalable authentication in a Node.js environment. It will focus on developing a comprehensive solution that incorporates modern authentication techniques such as JWT, OAuth2, and multi-factor authentication (MFA), ensuring that user data is protected while maintaining a seamless user experiences. Authentication ensures that only authorized users can access sensitive data or perform restricted actions. Without authentication, a web application is vulnerable to unauthorized access and malicious activities. Authentication allows the system to verify the identity of users, ensuring they are who they claim to be. This is crucial in preventing impersonation and identity theft.

Key Features of the System:

1.User Registration & Login: Users can sign up with credentials like email and password. Login allows users to authenticate and gain access.

2.Password Hashing & Security: Uses bcrypt or argon2 to hash passwords before storing them.Prevents plaintext password storage.

3. Token-Based Authentication (JWT): Uses JSON Web Tokens (JWT) for session management.

Literature Survey

User identification in Browser Environment via ML

User identification in browser environments via machine learning has gained traction due to its potential to enhance personalization and security. However, several limitations need to be addressed to ensure effective and ethical implementation.

Privacy Concerns:- Data Collection: Machine learning models often require extensive data, which can lead to privacy violations if sensitive information is collected without user consent.

Data Quality and Diversity:- Data Bias: Training data can be biased, leading to models that perform well on certain demographics but poorly on others.

Data Scarcity: High-quality labeled data is often source.[1]

A review of EEG based User Authentication trends and future research direction:

EEG-based user authentication leverages the unique electrical activity patterns of the brain to verify identities. This method promises high security and non-invasiveness but faces several limitations that need to be addressed in current research. Here's a review of these limitations and potential future research directions.

Inter-Session Variability: EEG signals can vary significantly across different sessions for the same user due to factors like fatigue, stress, and emotional state.

Inter-Subject Variability: There is high variability between individuals, making it challenging to develop a one-size-fits-all model.[2]

Enhancing JWT Authentication & Authorization in web Applications based on User Behavior History:

Enhancing JWT (JSON Web Token) authentication and authorization in web applications by incorporating user behavior history can significantly improve security and personalization. However, several existing limitations and challenges need to be addressed.

User Consent: Collecting and using behavioral data requires explicit user consent to comply with privacy regulations like GDPR and CCPA.

Data Sensitivity: Behavioral data is sensitive and must be handled with strict privacy and security measures to prevent misuse and breaches.

Data Accuracy: Inaccurate or noisy behavioral data can lead to incorrect assumptions about user behavior, impacting authentication and authorization decisions.

Authentication in modern web apps for data security using node. Js & Dark web:

Authentication in modern web applications for data security, particularly using Node.js and addressing issues related to the dark web, is a complex and multifaceted topic. Here are some key limitations of papers in this area, highlighting both technological and conceptual challenges:

Evolving Threat Landscape:

Emerging Threats: The dark web constantly evolves, with new threats and attack vectors emerging regularly. Research papers might quickly become outdated as new techniques and tools are developed by cybercriminals.

Adaptability: Methods proposed in papers might not be adaptable or robust enough to handle rapidly changing threats from the dark web.[4]

Implement advanced monitoring tools and analytics to detect and mitigate threats related to dark web activities:

Insufficient Coverage: Most monitoring tools do not cover the entire dark web, missing out on significant threat intelligence.

Lack of Real-Time Analysis: Delayed threat detection due to the absence of real-time monitoring capabilities.

Inadequate Threat Intelligence Integration: Limited integration with broader threat intelligence feeds reduces the effectiveness of threat correlation and analysis.[5]

A Frictionless secure user Authentication In web based Premium Applications:

Password-Based Authentication

User Fatigue: Repeatedly entering passwords can be tedious, especially for complex passwords. Security Risks: Passwords are vulnerable to phishing, brute force attacks, and poor user practices (e.g., password reuse).

Two-Factor Authentication (2FA)

User Inconvenience: 2FA often requires additional steps, such as entering codes from a mobile device, leading to frustration.

Dependence on External Devices: Loss of or failure to access the second device can lock users out of their accounts.[6]

A Password Based authentication system based on the CAPTCHA AI problem

Brute Force Attacks

Description: Attackers use automated tools to try many combinations of passwords.

Limitation: Simple CAPTCHAs can be bypassed by sophisticated bots, and complex CAPTCHAs can frustrate users.

Password Reuse and Weak Passwords

Description: Users often reuse passwords across multiple sites or choose weak passwords. Limitation: Existing systems rely heavily on user behavior, which is often inconsistent.[7]

A web Authentication Methods For web application

This paper reviews various web authentication methods used in web applications, highlighting their existing limitations. It aims to provide a comprehensive understanding of the challenges associated with these methods and propose potential solutions to enhance security and user experience.[8]

AI Driven Threat Detection

The rise of sophisticated cyber threats has necessitated the adoption of AI-driven threat detection systems. These systems leverage machine learning (ML) and artificial intelligence (AI) to identify patterns and anomalies indicative of malicious activities. While AI-driven threat detection offers significant advantages over traditional methods, it also presents unique challenges and limitations. This paper aims to critically analyze these limitations and suggest ways to overcome them.[9]

Authentication authorization in web services:

Role-Based Access Control (RBAC):

Static Roles: RBAC often lacks flexibility, as roles are predefined and may not cover all access scenarios.

Role Explosion: Managing a large number of roles can become complex and unmanageable.

Granularity: RBAC may not provide fine-grained control over permissions.

Attribute-Based Access Control (ABAC):

Complex Policies: ABAC policies can become highly complex and difficult to manage.

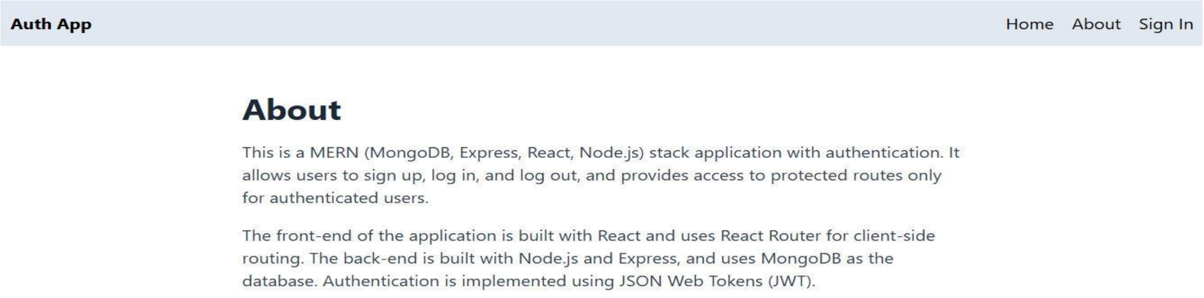
Performance Overhead: Evaluating multiple attributes for each access request can introduce performance issues.

Scalability: Scaling ABAC systems to large numbers of users and attributes can be challenging.[10]

Limitations Of Existing Work

- **Security Risks & Vulnerabilities:** Token Theft (JWT Expiration Issues).
- **Scalability Issues:** Session Management Overhead.
- **Complexity in Role-Based Access Control (RBAC):** Dynamic role changes.
- **User Experience Challenges:** Password Recovery Delays.
- **Dependency on Third-Party Libraries:** Passport.js Complexity.
- **Storage & Data Privacy Issues:** Token Storage Risks.
- **Logout & Token Revocation Issues:** Session Expiry Management.

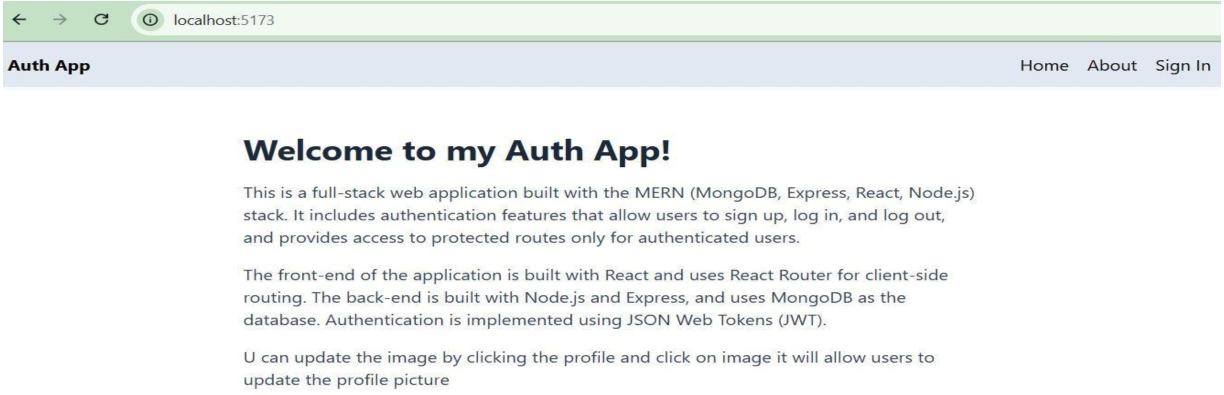
Results/ Outputs



About

This is a MERN (MongoDB, Express, React, Node.js) stack application with authentication. It allows users to sign up, log in, and log out, and provides access to protected routes only for authenticated users.

The front-end of the application is built with React and uses React Router for client-side routing. The back-end is built with Node.js and Express, and uses MongoDB as the database. Authentication is implemented using JSON Web Tokens (JWT).

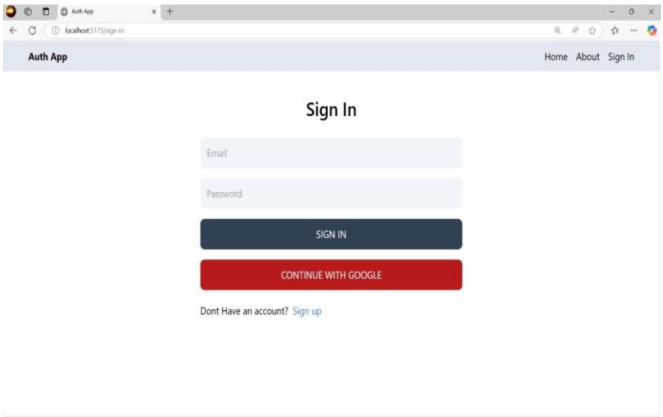


Welcome to my Auth App!

This is a full-stack web application built with the MERN (MongoDB, Express, React, Node.js) stack. It includes authentication features that allow users to sign up, log in, and log out, and provides access to protected routes only for authenticated users.

The front-end of the application is built with React and uses React Router for client-side routing. The back-end is built with Node.js and Express, and uses MongoDB as the database. Authentication is implemented using JSON Web Tokens (JWT).

U can update the image by clicking the profile and click on image it will allow users to update the profile picture



Sign In

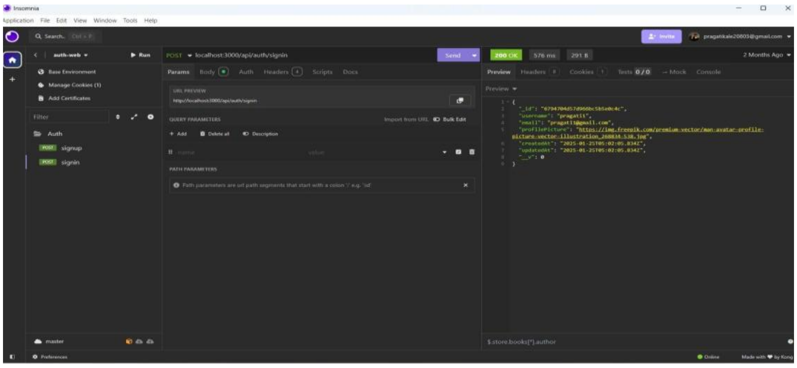
Email

Password

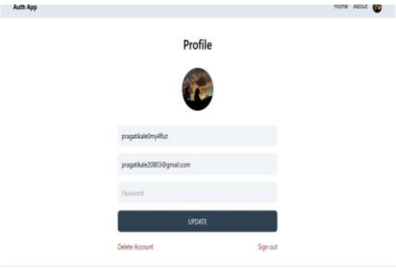
SIGN IN

CONTINUE WITH GOOGLE

Dont Have an account? Sign up


Insomnia
Application File Edit View Window Tools Help
Search.
Invite
pragatikale20803@gmail.com
auth-web
Run
Base Environment
Manage Cookies (1)
Add Certificates
Filter
Auth
POST signup
POST signin
master
Preferences
POST localhost:3000/api/auth/signin
Send
Params Body Auth Headers Scripts Docs
URL PREVIEW
http://localhost:3000/api/auth/signin
QUERY PARAMETERS
Import from URL Bulk Edit
Add Delete all Description
name value
PATH PARAMETERS
Path parameters are url path segments that start with a colon ':' e.g. ':id'
200 OK
576 ms
291 B
2 Months Ago
Preview Headers 8 Cookies 1 Tests 0 / 0 → Mock Console
Preview
$.store.books[*].author
Online
Made with ♥ by Kong

```
{
  "_id": "6794704d57d966bc5b5e0c4c",
  "username": "pragati1",
  "email": "pragati1@gmail.com",
  "profilePicture": "https://img.freepik.com/premium-vector/man-avatar-profile-picture-vector-illustration_268834-538.jpg",
  "createdAt": "2025-01-25T05:02:05.834Z",
  "updatedAt": "2025-01-25T05:02:05.834Z",
  "__v": 0
}
```



Profile

pragatikale0my4fut

pragatikale20803@gmail.com

Password

Conclusion

In this paper, we have explored and implemented various authentication mechanisms for web applications sing Node.js. The primary focus was on ensuring secure user authentication while balancing ease of use and performance. Future work could involve exploring additional security measures, such as multi-factor authentication and OAuth integration, to further enhance the application's security posture. Overall, this paper underscores the critical role of secure authentication in web applications and demonstrates how Node.js can be effectively employed to achieve this goal.

References

A. G. Uymin, O. A. Terentyeva // Bulletin of Cherepovets State University 2(119). 213–234 (2024). DOI 10.23859/1994-0637-2024-2-119-16. [CrossRef] [Google Scholar

M. K. Abdullah, K. S. Subari, J. L. C. Loong, and N. N. Ahmad, "Analy-sis of effective channel placement for an EEG-based biometric system,"

a.in Proc. IEEE EMBS Conf. Biomed. Eng. Sci. (IECBES), Nov. 2010, pp. 303–306, doi: 10.1109/IECBES.2010.5742249.

Beaudin, S.; Levy, Y.; Parrish, J.; Danet, T. An empirical study of authentication methods to secure e-learning system activities against impersonation fraud. Online J. Appl. Knowl.

Manag. 2016, 4, 42–61. [CrossRef] Paro, A., 17, M. A. | F., 03, B. S. | F., 28, J. S. | J., Richi Jennings | 4, M. V. | M., & 11, R. J. | M. (2021). Hackers leaked 22 million records on the dark web in 2020. | https://securityboulevard.com/2021/01/hackers-leaked-22-million-records-on-the-dark-web-in-2020/ [4

A.V. Vlasova, V. A. Dudarev, T. I. Novikova, Analysis Of The Principles Of The Systems Of Behavioral Analysis Of User Behavior And Entities //Fundamental and applied approaches to solving scientific problems. pp. 232–236 (2023)

P. Muthukrishnan, V. Sakthivel, B. Ramachandran, and K. Srihari, "Technical analysis on security realization in web services for e-business management," Inf. Syst. e-Bus. Manage., vol. 18, no. 3, pp. 427–438, Sep. 2020,doi: 10.1007/s10257-019-00423-w.

C. Meshram and M. S. Obaidat, "An efficient provably secure ibs tech-nique using integer factorization problem," in Proc. 1st Int. Conf. Comput.,Commun., Cyber-Secur., 2020, pp. 427–439.

Khan S. et al. ACM Computing Surveys 56. 6. 1–33 (2021)

A. G. Uymin, (2022) Control systems and information technologies. 2(88). 92–96

Bhargavan, Corin, Fournet, Gordon: Secure sessions for web services. ACM Transactions on Information and System Security (2007)