iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** II **Month of publication:** February 2024

DOI: https://doi.org/10.22214/ijraset.2024.58445

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue II Feb 2024- Available at www.ijraset.com

"QUICK" - Designing and Developing a Blog Application Using The MERN Stack

Jap Trusharkumar Patel
Parul University, Vadodara, Gujarat, India

Abstract: ***In today's digital age, blogging has emerged as a prevalent means of sharing ideas, experiences, and knowledge across the internet. I made "QUICK" using MERN (MongoDB, Express, React JS and Node JS. My Application not only provides the creation, editing, and deletion of blogs by authors but also build up user experience through advanced features such as categorization, filtering, and search functions. The main focus of this research is to explore how the MERN stack give to the seamless integration of front-end and back-end components. By leveraging React.js for the front end and Express.js and Node.js for the server-side logic, QUICK ensures a responsive and interactive user interface.***
Moreover, the implementation of features such as category-based organization, filtering options, and robust search, allowing readers to easily discover relevant content based on their preferences. Furthermore, it empowers authors with exclusive rights to edit and delete their own blogs.
Keywords: Mern stack, User experience, React.js

I. INTRODUCTION

Nowadays, blogging are very powerful tool for sharing ideas and experiences with world. As demand for user-friendly platforms grows, we introduce a novel solution: a MERN stack-based blogging platform – "QUICK", which not only empowers users to create captivating content but also enhances the overall blogging experience through intuitive features.

At the heart of our application lies a unique functionality designed to streamline the content creation process: automatic banner generation. Usually, bloggers are selecting and uploading a display image for their posts, a process that can be time-consuming and cumbersome. To enhance user convenience, It automates the banner selection process by fetching the first image when bloggers write their content in write blog page and seamlessly incorporating it as the banner for the blog post. This innovative feature not only reduces manual effort for bloggers but also ensures visually appealing and cohesive presentation across the platform.

In presenting this paper, I aspire to contribute valuable insights and best practices to the fields of web development, UI and UX design, and content management, advancing the discourse on innovative approaches to enhancing digital communication platforms. Through a synthesis of technical analysis, user feedback, and real-world application, we endeavour to illuminate the path toward creating more intuitive, efficient, and user-centric blogging experiences in the digital age.

II. LITERATURE REVIEW

The research paper explores the efficiency of an e-commerce web application built using the MERN stack along with modern tools like Tailwind CSS. It delves into the various aspects of ReactJS as a contemporary web development framework, highlighting its features and advantages in creating dynamic user interfaces. Additionally, the paper examines the utilization of Node.js for constructing a high-speed and scalable backend database server, focusing on its capabilities to handle concurrent requests efficiently. Moreover, it discusses the integration of Tailwind CSS, a utility-first CSS framework, and its impact on the development process and user interface design. By leveraging Tailwind CSS, developers can rapidly build and customize user interfaces with predefined utility classes, streamlining the styling process and promoting consistency across the application. Through the integration of these technologies and tools, the research aims to elucidate how they contribute synergistically to enhance the performance, scalability, and overall user experience of applications in the digital landscape. Furthermore, the paper provides a comprehensive analysis of the integration of these technologies and tools in web development, offering strategies and best practices for optimizing performance, scalability, and user experience. It also explores real-world case studies or examples where the MERN stack, ReactJS, Node.js, and Tailwind CSS have been successfully implemented in applications, providing empirical evidence and practical insights to inform developers, architects, and businesses seeking to leverage these technologies for building robust and competitive application solutions.

©IJRASET: All Rights are Reserved

Table 1. Literature Summary

Author	Title	Purpose
Prateek Rawat, *Archana N. Mahajan*	ReactJS: A Modern Web Development Framework	This paper delves into the exploration and analysis of ReactJS as a contemporary web development framework. Its purpose is to provide readers with an in-depth understanding of ReactJS's features, such as its component-based architecture, virtual DOM rendering, and state management capabilities.
C.M. K De Silva, *A. S De Silva,* *K.A. I Maduwantha,* *D.A.I.U Dewpura,* *D.I. De Silva* *And R.R.P De Zoysa6*	Efficiency of an E-Commerce Web Application with MERN Stack and Modern Tools	The purpose of this paper is to investigate and evaluate the efficiency of utilizing the MERN stack (MongoDB, Express.js, React.js, Node.js) along with modern tools in the development of an e-commerce web application.
S. L. Bangare1 *, S. Gupta2* *, M. Dalal3* *, A. Inamdar4*	Using Node.Js to Build High Speed and Scalable Backend Database Server	Explores the utilization of Node.js for constructing a high-speed and scalable backend database server, focusing on its asynchronous, event-driven architecture and capability to efficiently handle concurrent requests, vital for ensuring seamless scalability and performance.
Fadli Rifandi, *Tri Viqi Adriansyah,* *Rina Kurniawati*	Website Gallery Development Using Tailwind CSS Framework	The paper aims to provide insights into how Tailwind CSS can be leveraged to streamline the design and development process of website galleries, enhancing efficiency and facilitating the creation of visually appealing and responsive user interfaces.

III. METHODOLOGY

A. Design and implementation

1) Requirement Analysis and Planning:

a) Define the scope and objectives of the blogging application, outlining key features and functionalities required for a comprehensive user experience.

b) Plan the development process, including the selection of technologies, frameworks, and libraries to be used for frontend and backend development.

c) In development of QUICK I used MERN stack because Efficient Development Workflow, Scalability and Performance.

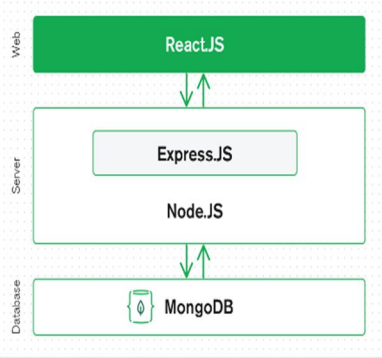


Figure 1. How does the MERN stack work?

2) Frontend Development with React.js and Tailwind CSS:

a) Design the user interface (UI) of QUICK using React.js, a JavaScript library for building interactive user interfaces.

b) Utilize Tailwind CSS, a utility-first CSS framework, for UI design to ensure consistency, responsiveness, and ease of styling.

c) Implement UI components for blog creation, editing, deletion, categorization, user authentication, and other application feature.

d) Integrate npm packages such as react-router-dom for client-side routing, Axios for making HTTP requests to the backend server, js-cookie for managing user authentication tokens, and react-quill for rich text editing in blog creation.

3) Backend Development with Express.js and MongoDB:

a) Develop the backend of application using Express.js, a web application framework for Node.js.

b) Set up MongoDB, a NoSQL database, to store blog posts, user information, and other application data.

c) Use Mongoose, an object modeling tool for MongoDB, to define data models, schema, and relationships between different entities.

d) Implement routes and controllers to handle CRUD operations for blog posts, user authentication, and other application functionalities.

e) Utilize npm packages such as cors for enabling cross-origin resource sharing, body-parser for parsing request bodies, bcryptjs for hashing passwords, and jsonwebtoken for generating and verifying JSON Web Tokens (JWT) for user authentication.

4) API Development and Integration:

a) Design and develop RESTful APIs to expose endpoints for creating, retrieving, updating, and deleting blog posts, as well as user registration and login.

b) Implement authentication middleware to verify user identity and authorize access to protected routes.

c) Integrate frontend and backend components of the application to ensure seamless communication and data exchange between the two layers.

d) Deployment and Optimization:

a) Deploy the blogging application to a hosting environment, such as Heroku, Netlify to make it accessible over the internet.

b) Implement optimization techniques, such as code minification, bundling, and caching, to improve the application's performance and load times.

c) Monitor application performance metrics, such as response times, server load, and error rates, and optimize as necessary to ensure scalability and reliability.

B. Main Functions of the Application

1) Automatic Banner Generation:

a) Implementation:

- Utilizing React.js and Tailwind CSS, the application implements a feature to automatically generate banners for blog posts by extracting the first image from the content.
- The functionality is integrated into the blog creation page, where users input their blog content.

The following code snippet demonstrates the implementation:
{extractFirstImgTag(blog.content).map((imgTag, index) => (
<div className='rounded-xl w-52 h-28 overflow-hidden' key={index}dangerouslySetInnerHTML={{ __html: imgTag }} />))}

b) Functionality:

- Upon submission of the blog content, the application parses the HTML content and extracts the first <img> tag encountered.
- If an image is found, it is displayed as the banner for the blog post.
- If no image is present, the banner remains empty

Extract First Image Tag Function:

```
const extractFirstImgTag = (htmlContent) => {
  const parser = new DOMParser();
  const doc = parser.parseFromString(htmlContent, 'text/html');
  const firstImgTag = doc.querySelector('img');
  return firstImgTag ? [firstImgTag.outerHTML] : [];
                                        };
```

IV. ARCHITECTURE DESIGN

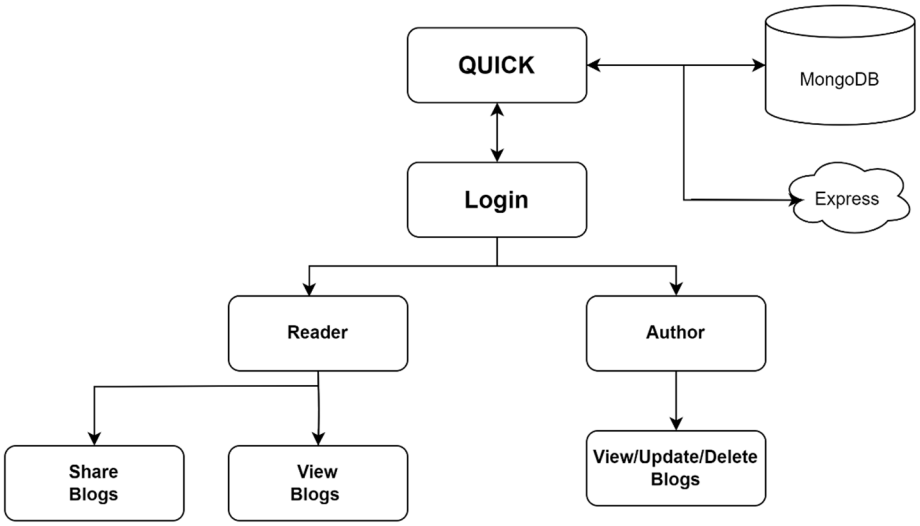


Figure 2. Architecture design of QUICK

In QUICK , The architecture is designed to provide a seamless experience for both readers and authors, with a focus on simplicity and efficiency. The application consists of three main components: login functionality for users, reader interface for consuming blog content, and author interface for creating and editing blog posts. Application always connected with MongoDB database and Express server.

1) Login Component:

- User Authentication: Implements secure user authentication mechanisms such as username/password authentication
- User Roles: Supports two primary user roles - reader and author. Each role has distinct permissions and access levels within the application.

2) Reader Interface:

- Blog Display: Presents a clean and intuitive interface for readers to browse through blog posts. Organizes blog posts in a categorized manner for easy navigation.

- Search Functionality: Enables readers to search for specific blog posts based on keywords, categories, or tags for efficient content discovery.
- Read-only Access: Restricts reader users from modifying or deleting blog content, ensuring the integrity of published posts.

3) Author Interface:

- Blog Creation: Provides a user-friendly interface for authors to create new blog posts. Includes text editor functionality with basic formatting options for content creation.
- Editing Capabilities: Allows authors to edit and update existing blog posts, including modifying text, adding media content, or changing post metadata.

V. RESULTS

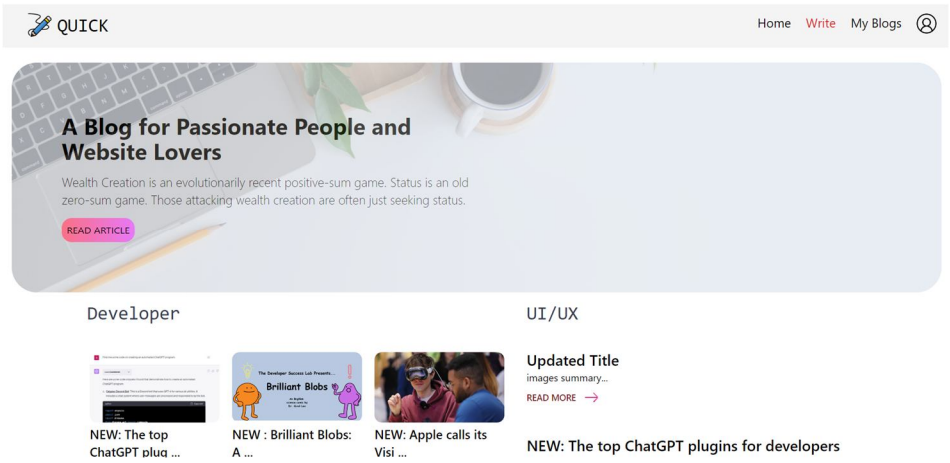


Figure 3. Home page

In Home page, Display All Author's Blog by different categories like Developer, Technology etc.
Moreover, it only shows banner image and title when reader click Read more button it move to next page where it read full blog.

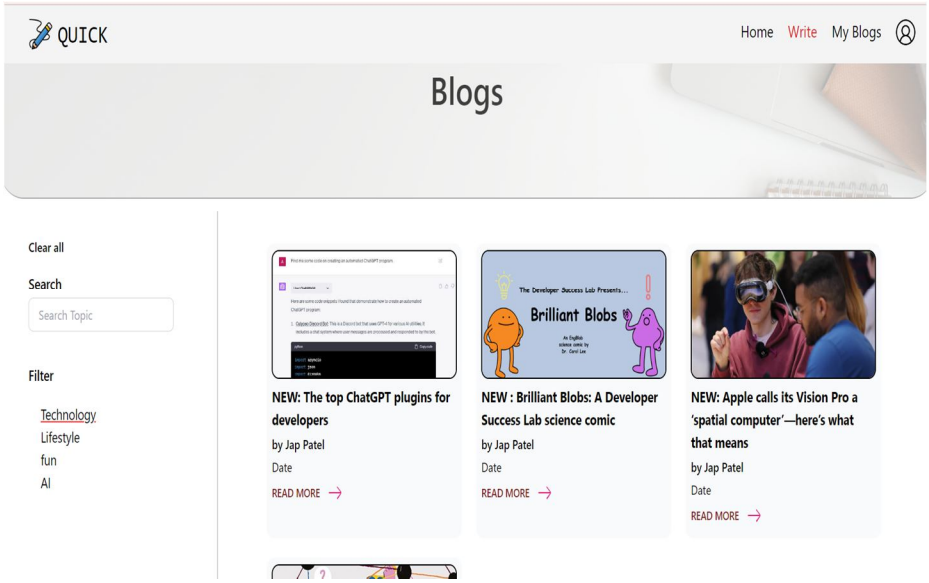


Figure 4. All blogs page

In this page, readers can find or filter blogs by categories through a user-friendly interface that allows them to browse or search for specific topics of interest. By selecting from a list of predefined categories or utilizing a search feature, readers can easily narrow down their focus to explore blog posts within their desired category.

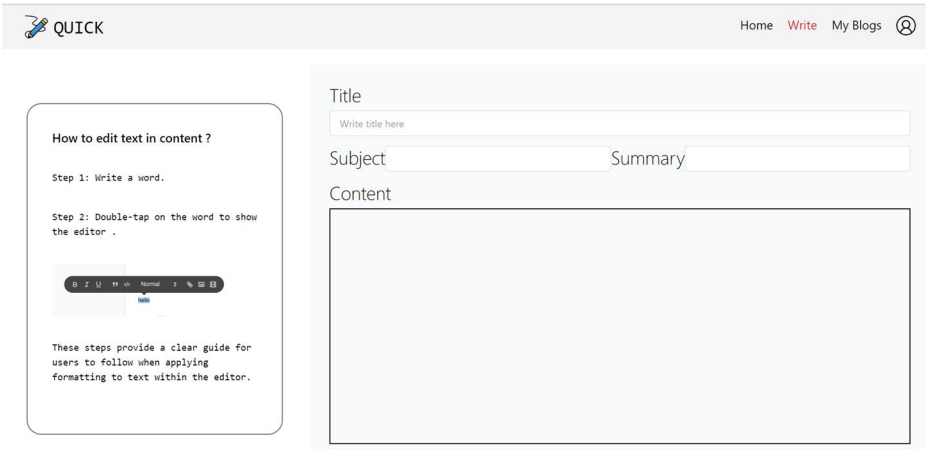


Figure 5. Write/Edit page

The "Write Blog Page" serves as a pivotal space where users can effortlessly craft and submit their blog content for publication. At its core, the page features an intuitive form interface, inviting users to input essential details such as the blog post title, category selection, and the body of their narrative

VI. CONCLUSION

In conclusion, the development of QUICK using the MERN stack has provided a comprehensive platform for creating, managing, and sharing blog content. It allows authors to write, edit, and delete their blogs seamlessly, enhancing their control over their published content. The inclusion of categorization, filtering, and search functionalities enhances user experience, enabling efficient navigation and discovery of relevant content. Through the implementation of features such as title, subject, summary, and content inputs, the website ensures that authors can effectively convey their ideas and engage their audience.

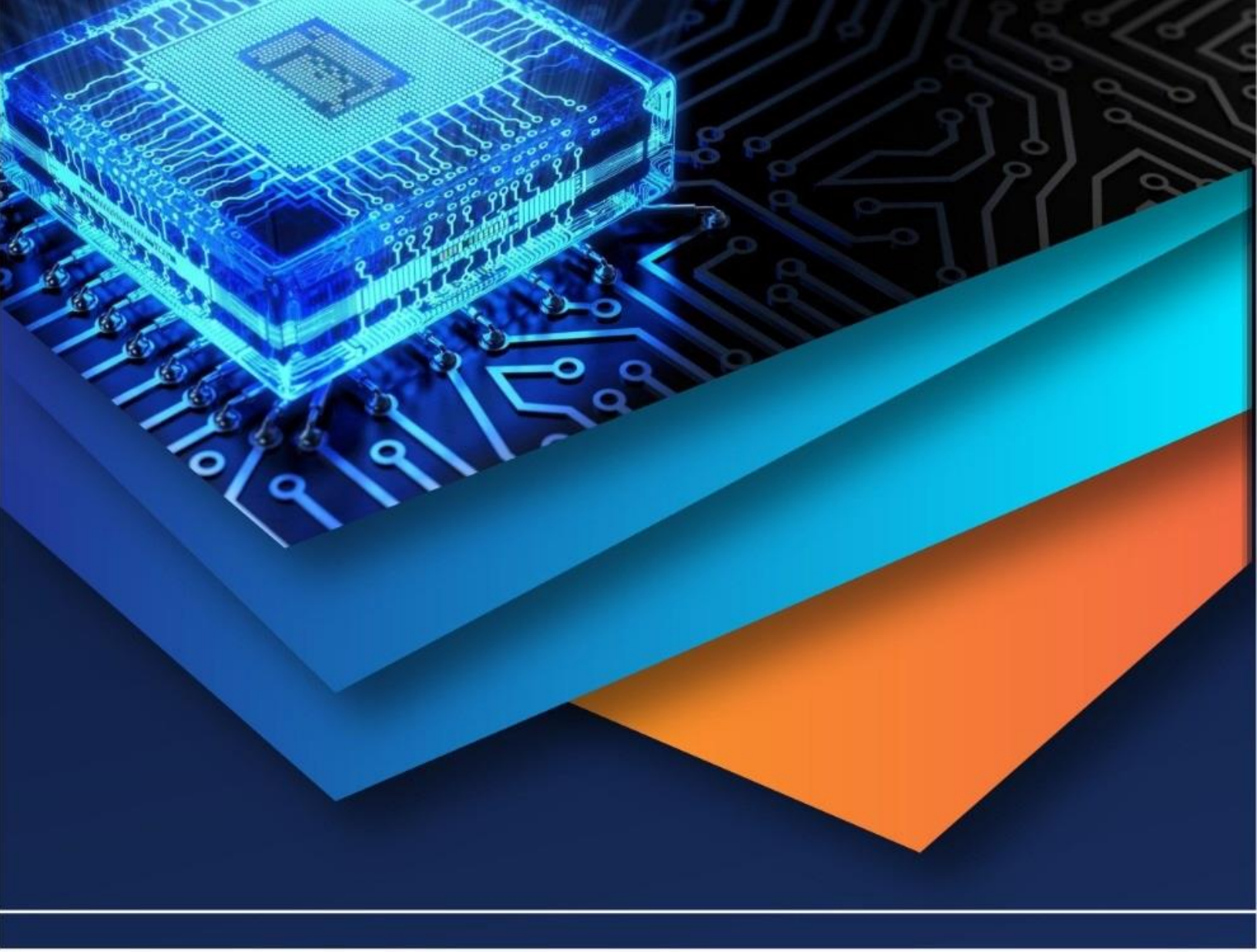
The project has not only demonstrated the technical proficiency in implementing a full-stack web application but also underscored the significance of user-centric design and functionality. By prioritizing features such as author-controlled blog management, content categorization, and search capabilities, the website caters to the needs and preferences of both authors and readers alike.

VII. ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to Mr. Akash for his guidance and support as our university guide.His mentorship has been invaluable in shaping our project and guiding us through its various stages. Additionally, we express our appreciation to Mr. Manoj for his mentorship and support as Internship mentor. His expertise and insights have greatly contributed to our learning and professional development.

REFERENCES

[1] Prateek Rawat , Archana N. Mahajan (Nov 2020). ReactJS: A Modern Web Development Framework. International Journal of Innovative Science and Research Technology, Volume 5, Issue 11.
[2] C.M. K De Silva, A. S De Silva, K.A. I Maduwantha, D.A.I.U Dewpura, D.I. De Silva and R.R.P De Zoysa (Oct 2022). Efficiency of an E-Commerce Web Application with MERN Stack and Modern Tools, International Journal of Engineering and Management Research, p349.
[3] S. L. Bangare , S. Gupta , M. Dalal , A. Inamdar (March 2016). Using Node.Js to Build High Speed and Scalable Backend Database Server. International Journal of Research in Advent Technology
[4] Fadli Rifandi , Tri Viqi Adriansyah, Rina Kurniawati (2022). Website Gallery Development Using Tailwind CSS Framework. EdsPoliteknik Piksi Ganesha Indonesia.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)