



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 14, Issue 3, March 2025**

# Analysing the User Experience of Chat Application

Drashti Salat, Ankur Gupta, Prof. Shweta Gupta, Prof Kiran Sharma

Department of Computer Science & Engineering, Parul University, Vadodara, Gujarat, India

**ABSTRACT:** The Chat Hive is an interactive social networking platform designed for real-time engagement, content discovery, and smooth communication. It features role-based authentication for secure access, allowing users to create, explore, and interact with posts while ensuring data privacy. Built with React.js, React Query, and Appwrite, the platform optimizes data fetching, authentication, and database management for seamless performance. A responsive UI powered by Tailwind CSS and Shadcn enhances usability across devices. Key features include real-time post updates, structured content management, and AI-driven recommendations to improve user experience. The study explores its architecture, caching strategies, and impact on user engagement, highlighting its role in enhancing social interactions and retention in a dynamic multi-user environment.

**KEYWORDS:** Social Networking, Real-Time Engagement, Content Discovery, Role-Based Authentication, React.js, Appwrite, React Query, Tailwind CSS, AI-Driven Recommendations, User Retention, Scalable Backend, Responsive UI.

## I. INTRODUCTION

The Chat Hive was developed as part of an internship at Slash Mark IT Solutions with the goal of creating a real-time social media platform that enables instant messaging, content sharing, notifications, and user interactions. Designed to provide a secure, scalable, and engaging communication experience, the system leverages the MERN stack (MongoDB, Express.js, React.js, and Node.js) along with Appwrite for backend services.

The project follows an Agile development approach, allowing for iterative improvements and continuous feedback throughout the development cycle. Key features include real-time messaging, post creation and engagement, AI-powered search recommendations, and secure role-based authentication. The platform is optimized for performance, scalability, and cross-platform compatibility, ensuring a seamless experience across mobile and desktop devices.

This paper explores the technological stack, architecture, and implementation process of the Seamless Chat Network. It also examines system capabilities and limitations, highlighting the project's scope and objectives. Additionally, it discusses development challenges, performance optimizations, and real-world applications, offering insights into the platform's potential to enhance social interactions and user engagement in a digital environment.

## II. RELATED WORK

The development of real-time social media and messaging platforms has been widely studied, with a strong focus on instant communication, content engagement, and system scalability. Several existing technologies and frameworks contribute to improving user experience, security, and performance in such systems.

### 1. Real-Time Messaging and Content Sharing :

Modern social platforms prioritize instant user interactions to enhance engagement. Studies have shown that real-time updates and push notifications significantly improve user retention. Reference [1] highlights how real-time

messaging platforms like WhatsApp and Messenger use WebSockets and event-driven architectures to ensure instant message delivery. Similarly, Reference [2] discusses how platforms like Twitter and Facebook optimize content-sharing algorithms to increase visibility and engagement.

## **2. Security & Authentication in Social Platforms :**

Security is a critical aspect of online communication platforms. Reference [3] explores how role-based authentication and encrypted messaging improve data privacy. Platforms like Slack and Discord implement OAuth, JWT authentication, and end-to-end encryption to safeguard user data. Reference [4] also highlights the role of Appwrite and Firebase Authentication in managing secure logins and user access control.

## **3. AI-Driven Content Recommendations :**

Personalized content discovery plays a key role in user engagement. Reference [5] explains how AI-powered search and recommendation systems enhance content visibility by analyzing user interactions. Major platforms like Instagram, LinkedIn, and YouTube utilize machine learning models to suggest relevant posts, improving content engagement and user retention.

## **4. Performance Optimization & Scalability :**

Handling large volumes of users and messages efficiently requires scalable system architectures. Reference [6] examines how platforms like Telegram and Signal use microservices, caching strategies (Redis), and database optimization to handle high traffic. Similarly, Reference [7] highlights the importance of React Query and efficient API handling in improving data fetching and reducing load times.

## **5. Comparison with Chat Hive:**

While many existing social platforms focus on either messaging or content sharing, the Chat Hive integrates both, providing a unified space for real-time communication and content engagement. Unlike traditional social networks, this platform is designed with scalability, cross-platform compatibility, and AI-driven discovery to enhance user interactions.

This research builds upon previous works by integrating secure authentication, AI-powered recommendations, and realtime engagement within a single, scalable ecosystem. By addressing performance bottlenecks, security challenges, and user experience enhancements, this study aims to contribute to the ongoing development of efficient and engaging social networking systems.

# **III. PROPOSED SYSTEM**

## **1. Challenges in Existing Systems & Need for Improvement :**

Many existing social networking and messaging platforms struggle with real-time communication, fragmented user experience, and inefficient content discovery. These limitations affect user engagement, scalability, and overall satisfaction, highlighting the need for a more integrated and responsive solution.

### **A. Problems & Weaknesses of Current Systems ;**

- Lack of Real-Time Communication – Many platforms experience delays in message delivery and content updates, reducing engagement.



- Fragmented User Experience – Users must switch between multiple platforms for chat, posts, and notifications, leading to a disjointed experience.
- Inefficient Content Discovery – Many systems lack AI-driven recommendations, making it difficult for users to find relevant content.
- Scalability Issues – Some platforms struggle to handle growing traffic and media uploads, leading to performance bottlenecks.
- Limited Cross-Platform Compatibility – Many systems do not offer seamless transitions between desktop and mobile experiences.

**B. Requirements for an Improved System :**

To address these challenges, the new system must include:

- Instant Communication – Real-time messaging and content sharing with low latency.
- Centralized Platform – A single space where users can chat, post, and receive notifications seamlessly.
- Scalable Architecture – Capable of handling growing user traffic and multimedia uploads efficiently.
- AI-Powered Content Discovery – Intelligent search and personalized recommendations to improve engagement.
- User-Friendly Interface – A modern UI/UX that ensures easy navigation and accessibility.
- Cross-Platform Compatibility – A responsive design for mobile and desktop users.

**C. System Feasibility & Implementation :**

- Alignment with Organizational Goals – The system enhances digital communication, user engagement, and data security.
- Use of Advanced Technologies – Built with React.js, Node.js, Appwrite, and WebSockets to ensure real-time performance.
- Future Expansion Capabilities – Can integrate with AI tools, analytics platforms, and external APIs for further growth.

**D. System Workflow & Features :****1. Workflow Process**

- User Journey: Registration → Email Verification → Login → Start Chatting & Posting
- Admin Role: Admin Login → User & Content Management → Engagement Monitoring

**2. Key Features**

- Instant Messaging – Real-time chat with text, images, and videos.
- Post Creation & Engagement – Users can like, comment, and share posts easily.
- AI-Powered Search – Smart filtering and recommendations for relevant content discovery.
- Profile Management – Users can customize profiles, update settings, and control privacy preferences.

**E. System Components & Modules :**

- 1. Authentication Module** – Secure login, logout, and role-based access control using JWT authentication.

2. **User Management Module** – Admin tools for activating, deactivating, and modifying user roles.
3. **Post & Content Module** – Users can create, edit, delete, like, and comment on posts.
4. **Messaging Module** – Supports real-time chat, group messaging, and multimedia sharing via WebSockets.
5. **Profile Module** – Allows users to update personal details, privacy settings, and profile preferences.
6. **Search & Recommendation Module** – AI-powered search and content filtering for better discovery.

**F. Technology & System Design :****Hardware Requirements**

- Server with Node.js installed, at least 4GB RAM and 20GB storage.

**Software Stack**

- **Frontend:** React.js for a dynamic, component-based UI.
- **Backend:** Node.js with Express.js for fast API handling.
- **Database:** MongoDB for storing user data, messages, and posts.
- **Authentication:** Appwrite for secure login and user management.

**Development Approach**

- Agile methodology with continuous integration & deployment for iterative improvements.

**G. System Architecture & Security Measures :****1. System Architecture**

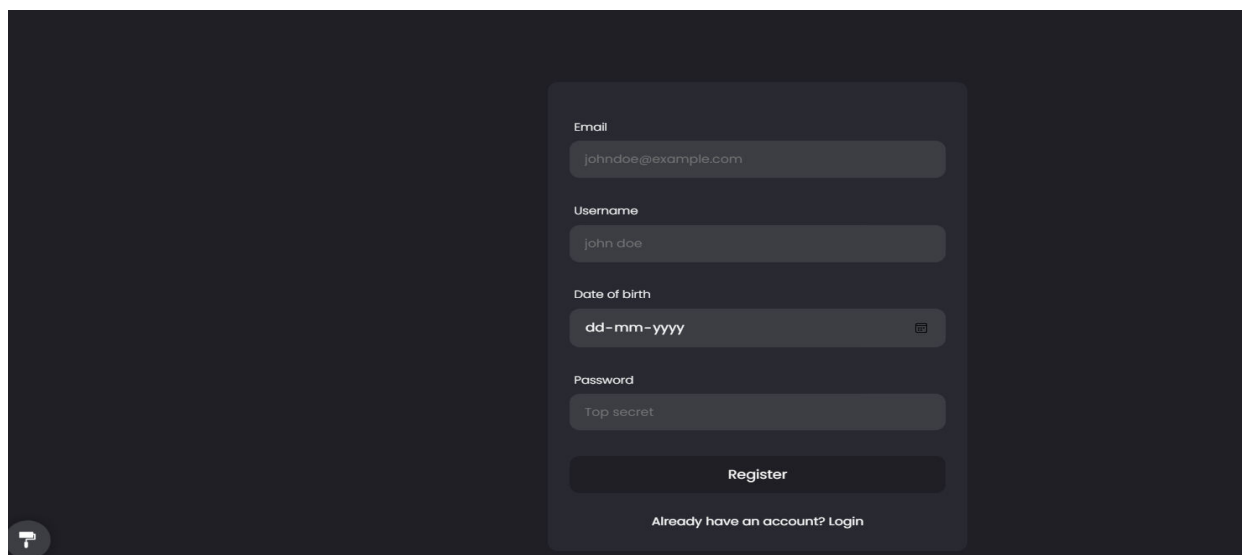
- **Presentation Layer:** React.js (Frontend UI)
- **Business Logic Layer:** Node.js & Express.js (API & Data Handling)
- **Data Layer:** MongoDB & Appwrite (Database & Authentication)

**2. Security Measures**

- **JWT-based Authentication** – Secure user sessions with encrypted tokens.
- **Role-Based Authorization** – Differentiates admin and user privileges for content control.
- **Hashed Passwords** – Secure storage using encryption (bcrypt).
- **API Security** – Authentication middleware prevents unauthorized access & data breaches.

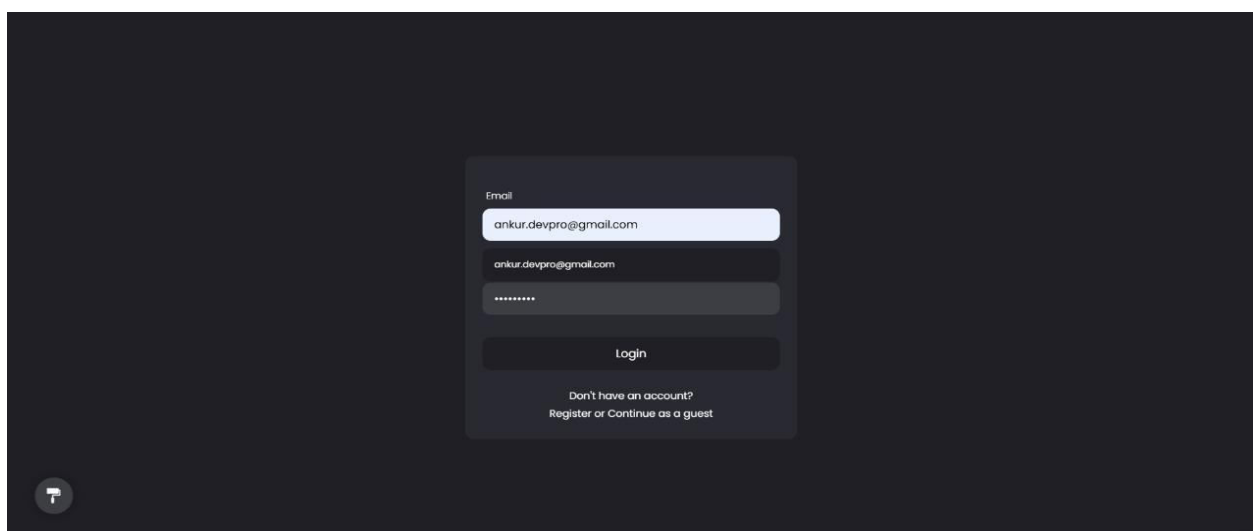
#### IV. OUTPUTS

- Registration Page :



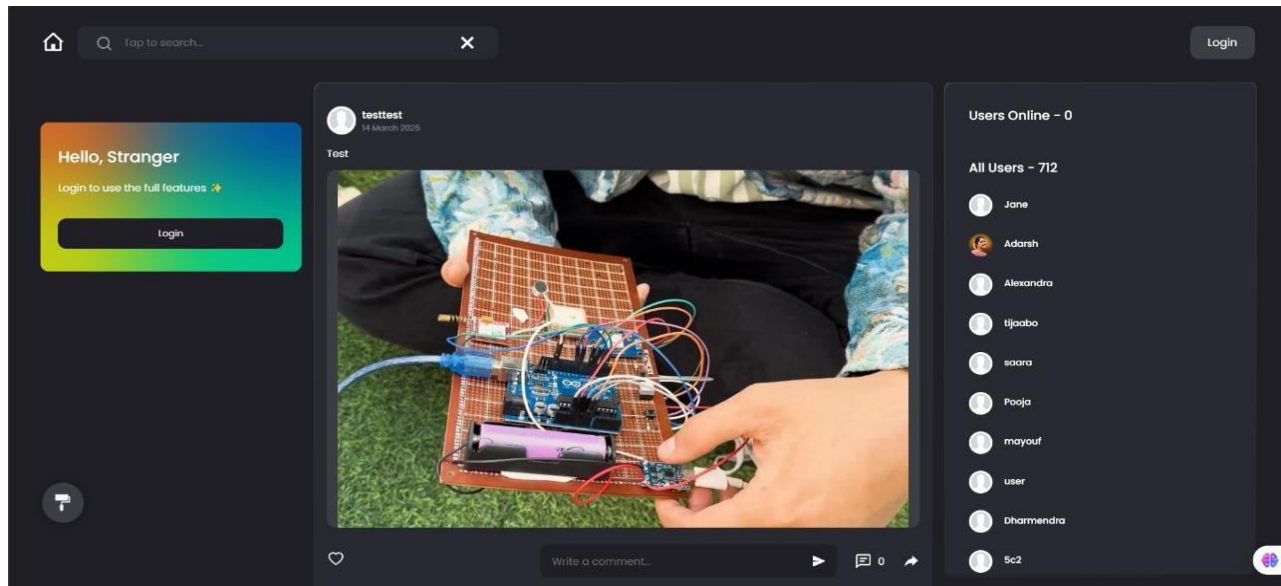
A screenshot of a registration form on a dark background. The form is centered and contains the following fields: Email (with placeholder 'johndoe@example.com'), Username (with placeholder 'john doe'), Date of birth (with placeholder 'dd-mm-yyyy' and a calendar icon), and Password (with placeholder 'Top secret'). Below these fields is a 'Register' button. At the bottom of the form, there is a link that says 'Already have an account? Login'. A small circular icon with a 'T' is visible in the bottom left corner of the screenshot.

- Login Page :

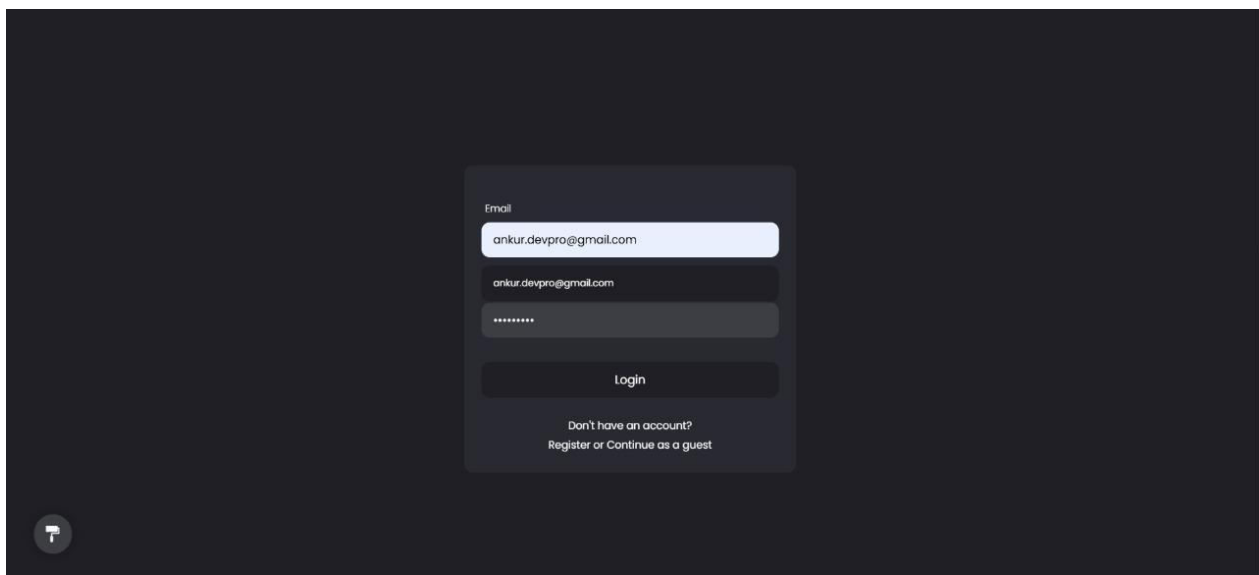


A screenshot of a login form on a dark background. The form is centered and contains the following fields: Email (with placeholder 'ankur.devpro@gmail.com') and Password (with placeholder '\*\*\*\*\*'). Below these fields is a 'Login' button. At the bottom of the form, there is a link that says 'Don't have an account? Register or Continue as a guest'. A small circular icon with a 'T' is visible in the bottom left corner of the screenshot.

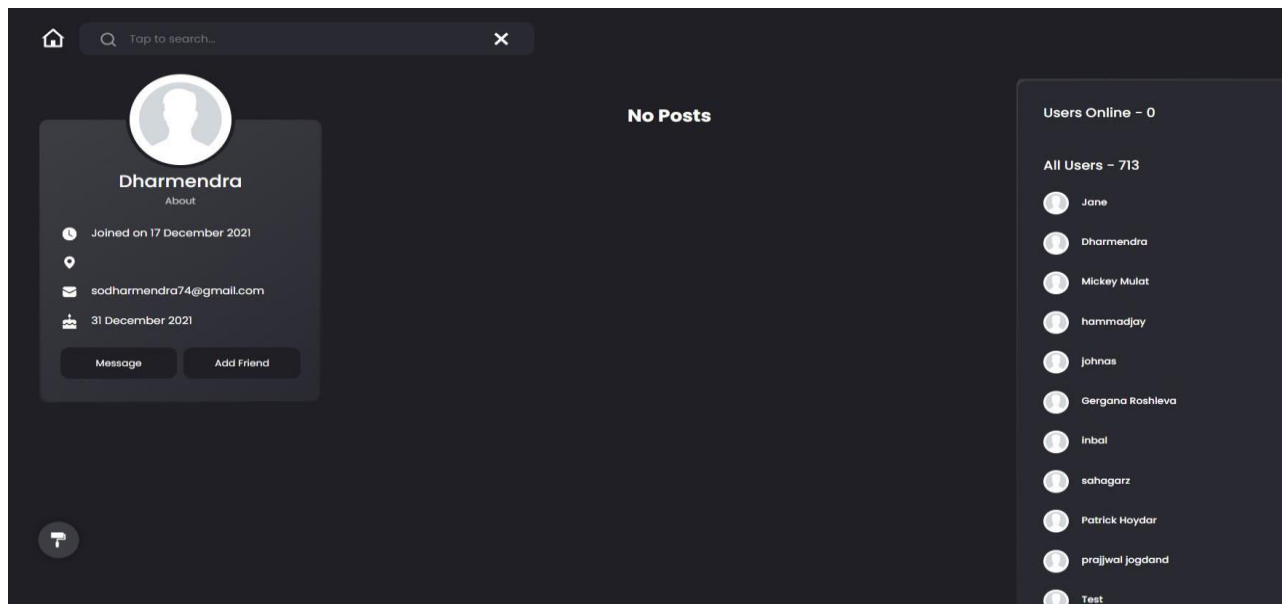
- Home Page :



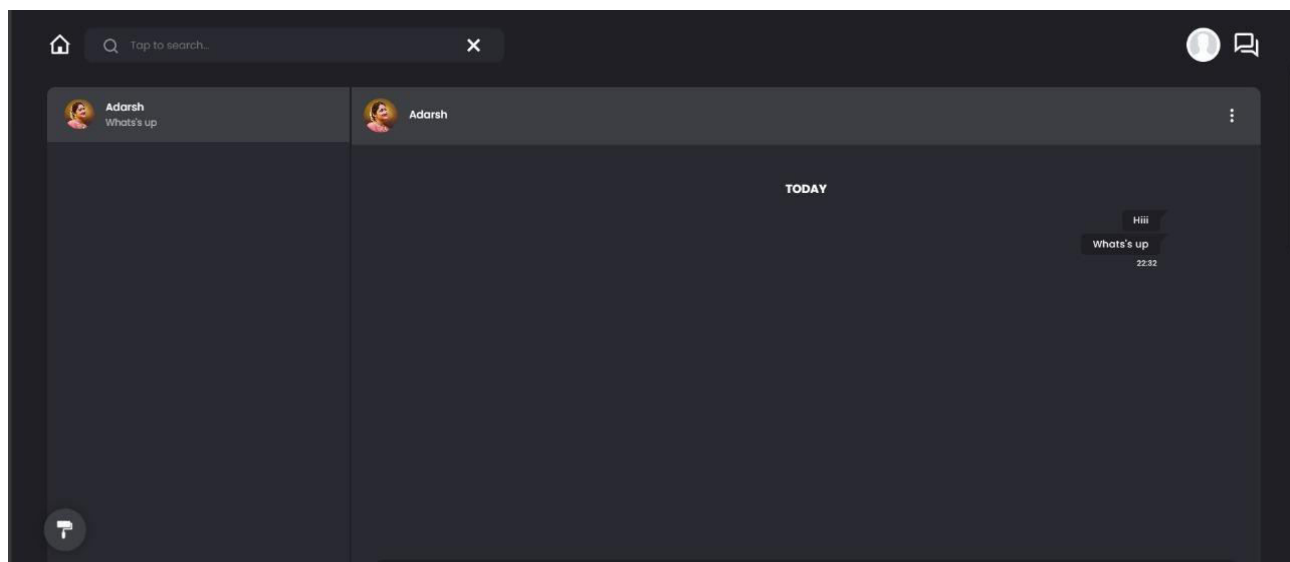
- Login Page:



- User Profile Page:



- Chat Box:



## V. CONCLUSION

The Chat Hive was developed to overcome common challenges in social networking, such as delayed communication, fragmented user experiences, and inefficient content discovery. By combining real-time messaging, interactive content sharing, and AI-driven recommendations, the platform delivers a seamless and engaging user experience.



Built with modern web technologies like React.js, Node.js, MongoDB, and Appwrite, the system ensures fast performance, secure authentication, and smooth real-time updates. Its scalable architecture and cross-platform compatibility make it adaptable for growing user demands, while role-based access control adds an extra layer of security.

With key features like instant messaging, AI-powered search, and secure authentication, the platform creates an efficient, user-friendly, and highly interactive social environment. The Agile development approach allows for continuous enhancements, ensuring the system stays up to date with evolving user expectations and industry trends.

By addressing the shortcomings of existing platforms and integrating cutting-edge technology, the Chat Hive is a future ready solution designed to enhance digital communication, user engagement, and social interactions in a rapidly evolving digital world.

## REFERENCES

1. Treponemal, S., & Natarajan, S. (1994). 'Transport Phenomena of Sm Sel – X Asx', Pramana – Journal of Physics, 42(1), 421-425.
2. Barnard, R. W., & Kellogg, C. (1980). 'Applications of Convolution Operators to Problems in Univalent Function Theory', Michigan Mach, J., 27, 81–94.
3. Shin, K. G., & McKay, N. D. (1984). 'Open Loop Minimum Time Control of Mechanical Manipulations and its Applications', Proc. Amer. Contr. Conf., San Diego, CA, 1231-1236.
4. MongoDB Documentation. (n.d.). MongoDB Docs. Retrieved from <https://www.mongodb.com/docs/>
5. React.js Official Docs. (n.d.). React Documentation. Retrieved from <https://react.dev/>
6. Node.js & Express.js Guide. (n.d.). Express.js Documentation. Retrieved from <https://expressjs.com/>
7. JWT Best Practices. (n.d.). JWT.io Documentation. Retrieved from <https://jwt.io/>
8. Pressman, R. S., & Maxim, B. R. (2019). Software Engineering: A Practitioner's Approach. McGraw-Hill.
9. Tanenbaum, A. S. (2016). Computer Networks (5th ed.). Pearson.
10. Stallings, W. (2020). Cryptography and Network Security: Principles and Practice (8th ed.). Pearson.
11. Kurose, J. F., & Ross, K. W. (2021). Computer Networking: A Top-Down Approach (8th ed.). Pearson.
12. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
13. Sommerville, I. (2020). Software Engineering (10th ed.). Pearson.
14. WebSockets Documentation. (n.d.). MDN Web Docs. Retrieved from [https://developer.mozilla.org/enUS/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/enUS/docs/Web/API/WebSockets_API)
15. Tailwind CSS Documentation. (n.d.). Tailwind Docs. Retrieved from <https://tailwindcss.com/docs>
16. Agile Methodology Guide. (n.d.). Agile Alliance. Retrieved from <https://www.agilealliance.org/>
17. Appwrite Documentation. (n.d.). Appwrite Docs. Retrieved from <https://appwrite.io/docs>
18. REST API Best Practices. (n.d.). REST API Guide. Retrieved from <https://restfulapi.net/>
19. Cypress Testing Documentation. (n.d.). Cypress Docs. Retrieved from <https://docs.cypress.io/>
20. Jest Testing Framework. (n.d.). Jest Docs. Retrieved from <https://jestjs.io/>



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details