Amity Journal of Computational Sciences
ISSN 2456-6616 (Online)
Volume 7, Issue 1, 2023

Chat Application using React.js and Firebase

Rishika Pandey
Undergraduate Scholar,
Greater Noida, Uttar Pradesh, India

Abstract

Communication and engagement have been transformed by the development of web technology, ushering in a new era of real-time connectedness. This study investigates the creation of a cutting-edge real-time chat application using Firebase, a flexible platform with real-time database capabilities, and React.js, a well-known JavaScript toolkit for creating user interfaces. The objective of developing this application is construction of an effective, scalable, and user-friendly chat application is the main objective of this study, with a focus on the smooth integration of React.js and Firebase to enable real-time communication.

I. INTRODUCTION

In the contemporary digital landscape, communication is no longer confined to traditional mediums; it has evolved into a real-time, dynamic exchange facilitated by web applications. Instant messaging and real-time chat applications have become integral components of our daily lives, transforming the way we connect and interact. This paradigm shift has been fuelled by advancements in web technologies, enabling the creation of sophisticated, responsive, and user-friendly applications that emulate real-life conversations [1].

This study is significant because it has the potential to enrich the ecosystem of real-time communication applications, which is constantly growing. This research aims to motivate and instruct developers in creating effective and interesting chat applications by exhibiting the possibilities that result from the combination of React.js with Firebase. Insights gathered from the performance assessment and user testing will also open the door for additional improvements and innovations, addressing the changing needs of a user base.

A. Historical Perspective

Real-time communication online dates back to the early days of the internet and the emergence of the internet. An important milestone was reached with the introduction of Internet Relay Chat (IRC) in the late 1980s and early 1990s, which gave people all around the world a platform for real-time text-based communication [3]. From the AIM and Yahoo Messenger of the late 1990s to the contemporary behemoths like WhatsApp, Facebook Messenger, and Slack, many chat systems have developed over time, each contributing to the advancement of real-time communication.

B. Advantages of chat applications:

Because of its many benefits, chat programs are widely used in both personal and professional situations. Important benefits include:

Instant Communication: Chat programs make it possible to send and receive messages instantly, promoting in-person conversations that are not limited by distance.

Global Connectivity: Users have the ability to connect with people or organizations all around the world, creating global collaboration and a sense of a linked world [2].

Rich Media Sharing: New chat applications make it possible to share a variety of media types, such as pictures, videos, documents, and voice messages, fostering cooperation and communication.

Effective Information Flow: Chat applications speed up the dissemination of information, improving productivity and the ability of teams and organizations to make decisions.

C. Issues with Chat Applications

However, the widespread usage of chat programs comes with a unique set of problems:

Security and privacy: Protecting user information and maintaining privacy in chat applications is a major challenge. Secure authentication methods and encryption are crucial.

Scalability: As user bases expand, it becomes increasingly difficult to ensure that the chat program can scale to accommodate the increased traffic and message volume.

Maintaining Real-time Communication: A technical challenge that requires effective solution is achieving true real-time communication with minimal latency and great dependability.

User Experience and Interface Design: To guarantee user happiness and engagement, it is essential to strike a balance between a feature-rich interface and an intuitive, user-friendly design [4].

In this study, we concentrate on utilizing the capabilities of Firebase, a comprehensive cloud-based platform, and React.js, a widely used JavaScript library, to build a real-time

chat application. The combination of these technologies intends to take use of React.js' prowess in creating dynamic user interfaces and Firebase's real-time data synchronization skills. The ultimate objective is to develop a real-time chat application that is effective, scalable, and engaging that responds to the changing communication requirements of contemporary society. Additional sections will go into the built chat application's technical specifics, architecture, design, implementation, and evaluation, providing insightful conclusions and highlighting potential future developments in this field.

II. LITERATURE REVIEW

A. Overview of React.js

React.js, sometimes known as React, is a well-known open-source JavaScript library that is supported by Facebook as well as a community of independent programmers and businesses [5]. Due to its distinctive method of creating user interfaces (UIs) for online apps, it was initially introduced in 2013 and has since acquired enormous traction.

Reusable components are the foundation of React.js' design. A React component is a self-contained, reusable bit of code that contains a particular UI element and its functionality. These elements can be combined to build intricate user interfaces, facilitating efficient and maintainable development [6]. The component-based architecture of the library enables programmers to divide the user interface into smaller, more manageable chunks, making it simpler to reason about and debug.

The usage of a virtual DOM (Document Object Model) is one of React's distinguishing characteristics. React builds a virtual version of the DOM in memory rather than modifying the real DOM directly [7]. React determines the most effective approach to update the actual DOM when changes are made by comparing the virtual DOM with the previous iteration. This method reduces direct DOM interaction to a minimum, improving performance and enhancing user interface responsiveness.

Additionally, React.js supports declarative programming, enabling developers to specify the intended UI state based on the data being used at the time. React eliminates the need for imperative DOM manipulation by updating the UI automatically when the data changes.

B. Overview of Firebase and Its Features

Google purchased Firebase, a complete platform for creating mobile and online applications, in 2014. It offers a wide range of resources to make it easier to create feature-rich, scalable, and secure applications [10]. A wide range of features required for contemporary app development are provided by Firebase, enabling developers to create applications fast and effectively.

Real-time data synchronization across all connected clients is made possible by Firebase's NoSQL, cloud-hosted database, which is one of the platform's key features. For programs like chat programs that demand real-time updates, this is essential.

Authentication: Developers can easily authenticate users using a variety of techniques, including email addresses and passwords, phone numbers, social identity providers (such as Google, Facebook, and Twitter), and more, thanks to Firebase's simple-to-integrate authentication services [11].

Firebase Cloud Functions give programmers the ability to execute backend code in response to events brought on by Firebase features and HTTPS requests. Event-driven architectures and serverless computing are now possible.

Hosting: With features like SSL support, global content delivery networks (CDN), and continuous deployment, Firebase Hosting provides a safe and simple method to host online apps [12].

Storage: For storing and serving user-generated material, including as photographs, videos, and files, Firebase Storage provides a scalable object storage solution.

Cloud Messaging: Developers may send dependable and effective push alerts to their applications using Firebase Cloud Messaging (FCM), which keeps users interested and informed [13].

Firebase's integration with Google's machine learning tools enables developers to use machine learning models in their apps without having any prior knowledge in the field.

Firebase Crashlytics With the help of real-time monitoring and comprehensive crash reports from Crashlytics, developers can quickly find and address problems with their applications.

C. Previous research and initiatives pertaining to chat applications

In the fields of computer science and software engineering, the creation of chat apps has been a continuing topic of interest. Many studies and initiatives have investigated numerous chat application characteristics, concentrating on improving user experience, maximizing performance, and addressing security issues.

1. Case Study on WhatsApp End-to-End Encryption [1]

This study explores WhatsApp's security procedures, one of the most widely used messaging platforms worldwide [14]. It examines the end-to-end encryption technologies used by WhatsApp to guarantee the confidentiality and security of user communications, illuminating the developments in real-time messaging security.

WhatsApp enables its users to stay in touch with friends and family. In addition to encouraging its users to connect with one another and stay in touch, it also encourages them to form groups and communicate images, videos, recordings, documents, archives, and audios [15]. The importance of ensuring its users' privacy and security has become more and more important as an ever-growing number of people use

WhatsApp as a way of mobile communication. The application's users expect a reasonable level of privacy protection for each and every one of their communications..

2. Design and Implementation of Web Based real time chat interfacing server

Instead of only relying on outside resources, this research paper attempts to make Indonesia self-reliant in the development of technology, especially chat applications. Indonesia still lacks a locally created chat program despite the abundance of third-party chat software. Building a native chat application to fill this market vacuum and meet the demand for a domestic communication platform is what spurred the effort.

III. METHODOLOGY:

The methodology used in this project involves a thorough comprehension and integration of React.js and Firebase in order to build a productive real-time chat application. The following crucial elements are part of the development process:

A. An Overview of Firebase's Real-Time Database Features

Google created Firebase, a comprehensive cloud-based platform that provides a variety of tools and services for creating online and mobile applications. It has attributes including authentication, a live database, cloud functionality, and more [17].

Synchronization of Data in Real Time: A NoSQL database hosted in the cloud, Firebase Realtime Database enables real-time data synchronization. It is perfect for real-time applications like a chat application since it enables multiple clients to receive updates practically instantly when data changes.

B. React.js and Firebase integration for a real-time chat application

Architectural Integration: Building a solid system architecture that makes use of both technologies' advantages is necessary to integrate React.js and Firebase. The user interface is handled by React.js, and the backend operations and real-time data synchronization are handled by Firebase.

Handling Real-time Messaging: Using Firebase to provide real-time messaging features within the application ensures that messages are delivered and updated instantly for all users participating in the discussion.

IV. System Architecture and Design:

1. Frontend (Client-side) architecture

React component:

- A well-liked JavaScript library for creating interactive user interfaces (UIs) is ReactJS.
- Each component has a specific function and encompasses a section of the UI, fostering maintainability and reusability.
- The "Chat", "Messages", "Input", "Navbar", "Search", and "Chats" components, for instance, are essential elements of the application's user interface.

Hierarchy of Components:

- The hierarchy of the component organization reflects the UI's organizational structure.
- The state and behaviour of a child component is controlled by the parent component.
- The "Chat" component in the program manages the "Messages" and "Input" child components as child components.

State Management:

- The application's state is managed using React's local state and Context API.
- While the Context API enables state to be shared across several components, local state stores information that is unique to a component.
- Tracking crucial data including current user data, messages, and conversation history requires state management.

Routing:

- The application uses the well-known routing library React Router for navigation.
- Different URLs are associated with various components, allowing for easier navigation and improved user experience.

2. Server-side (Backend) Architecture (Firebase)

Services for Firebase:

- Backend services for web and mobile applications are provided by Firebase, an extensive cloud-based platform.
- The application makes use of several important Firebase features, including Firebase Authentication, Firestore , and Storage.

Database in real-time (firestore):

- Real-time data synchronization between devices is possible thanks to Firestore, a NoSQL cloud database.
- Users are updated in real-time thanks to the storage of chat data, messages, and personal information.

Firebase Auth authentication

- Secure user authentication is handled by Firebase Authentication utilizing a number of techniques, including email and password.
- It maintains a safe login process by managing user IDs and restricting access to the application.

Firebase Storage in the Cloud:

For safe file uploads and downloads, such as avatars and photographs, Firebase Storage is used, expanding the user experience with multimedia features.

3. Integration

API Calls:

- Through Firebase SDKs and APIs, the frontend communicates with Firebase services.
- To ensure efficient communication with the backend, API calls are used to carry out tasks like sending messages, obtaining chats, and maintaining user authentication.

Real-time Updates:

- The real-time features of Firebase are used to give the application immediate updates when data in Firestore changes.
- Users can view fresh messages and other updates in real-time, which improves the application's responsiveness and interaction.

V. Implementation:

1. Environment Setup and Project Initialization:

Setting up the development environment is the first step in starting the Real-Time Chat Application project. If they are not already installed, download and install yarn, node.js, and npm (Node Package Manager). These programs are necessary for running JavaScript programs and handling dependencies. Create React App should then be used to launch a fresh ReactJS application. Use the npx create-react-app real-time-chat-app command to start the app. This creates the foundational files and directories for a React application.

2. Firebase Project and Configuration:

Project & Configuration for Firebase: Integrate the potent cloud-based platform Firebase to manage backend services. Create a Firebase project in the Firebase Console to get started. Set up the Firebase project settings to configure the authentication, firestore, and storage services. Setting up a real-time database (Firestore) for data storage, a cloud storage system (Storage), and user authentication are all required. Obtain the Firebase configuration credentials needed to connect the React application to Firebase, including apiKey, authDomain, projectId, storageBucket, messagingSenderId, and appId.

3. Client-Side (Frontend) Development:

Project & Configuration for Firebase: Integrate the potent cloud-based platform Firebase to manage backend services. Create a Firebase project in the Firebase Console to get started. Set up the Firebase project settings to configure the authentication, firestore, and storage services. Setting up a real-time database (Firestore) for data storage, a cloud storage system (Storage), and user authentication are all required. Obtain the Firebase configuration credentials needed to connect the React application to Firebase, including apiKey, authDomain, projectId, storageBucket, messagingSenderId, and appId.

4. Component Functionality and State Management:

State management and component functionality: Describe each component's capabilities. While user input and message sending are handled by the Input component, message display is handled by the Messages component. Utilize the Context API and local state of React for effective state management. While the Context API enables state exchange between distinct components, local state stores data that is unique to a component. The management of important data, including the data of the current user, message history, and chat information, depends heavily on state.

5. Routing and Navigation:

Using React Router, implement routing to make it possible for users to navigate the application with ease. To ensure that the correct components are rendered based on the URL, different URLs will map to different components. The smooth and simple user experience that users get when navigating the program depends on routing.

6. Deployment:

The client-side (frontend) of the application must be constructed and functional before it can be deployed to a hosting platform and made available to users. There are several deployment alternatives, but let's focus on using Firebase Hosting for this project because of its smooth connection with the Firebase backend.

- Setup for Firebase Hosting:

Start by using npm to install the Firebase CLI (Command Line Interface) globally. Npm install -g firebase-tools should be used. After that, log in to Firebase using the firebase login. Use firebase init to set up Firebase hosting in the root directory of the React application. To set up the hosting settings, adhere to the directions. A public directory and a firebase.json file will be created as a result.

- Build and Deploy:

Use the npm run build command to create the React application. In the build directory, this compiles the program into optimized, production-ready code. Use firebase deploy to deploy. The build directory's contents will be uploaded, and Firebase Hosting will offer a public URL where the application is hosted.

- Accessing the Deployed Application:

The deployed application can be accessed using the given Firebase Hosting URL when deployment has been completed. The Real-Time Chat Application is now accessible to users by going to this URL in their web browser.

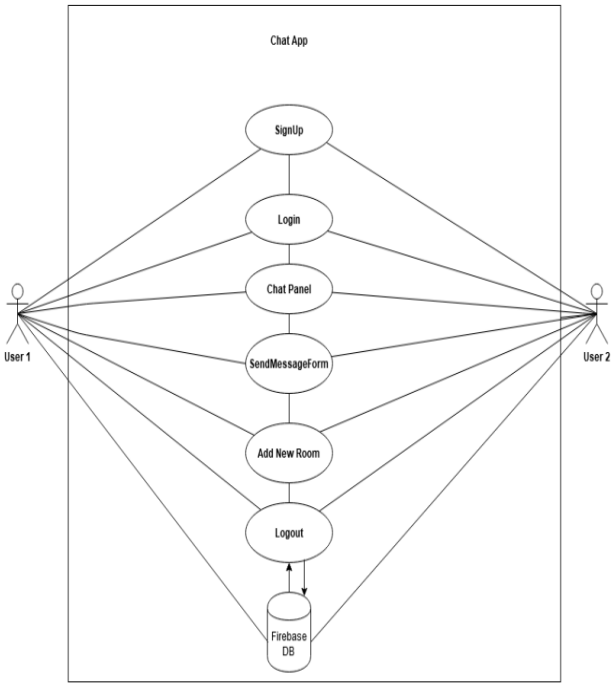

Fig: Use case Diagram

Users can access the Real-Time Chat Application through a web browser on any operating system by entering a specific URL.

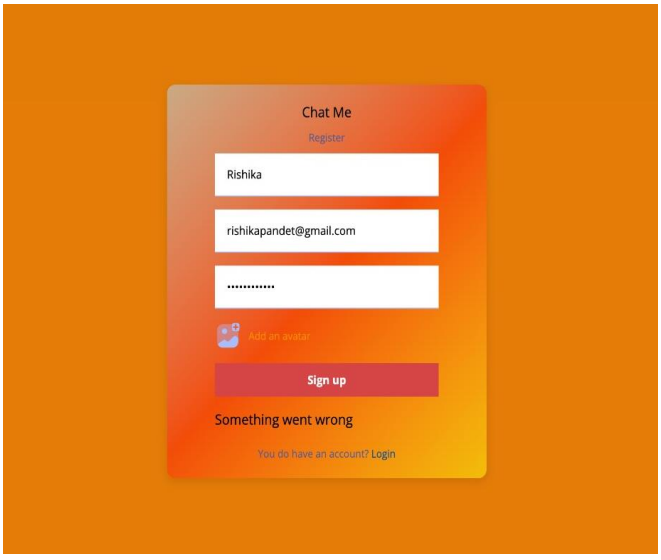

Fig: Screenshot of Sign-up page

Users must have a working email address and a six-character password to sign in. Users are able to sign up successfully and log in to the app to begin communicating. Pressing enter sends messages that have been written in the message bar at the bottom of the panel. In the chat window, the sent messages show the sender's email address and a timestamp. In the same panel, recipients can view and respond to messages.

Fig: Screenshot of Login Page

The backend services provided by Firebase are crucial to the operation of the application, which uses Firebase to host its database and store real-time data. A user's chat session can be ended by simply logging out from within the chat window, making for a full and engaging conversing experience.

To create user IDs with email and password authentication, utilize Firebase's auth() function. The user's email address and password, which are normally received from a registration form, are collected when this function is called. This data is then sent securely to Firebase Authentication services.

The auth() function starts a process that entails encrypting and safely storing this information within Firebase Authentication once it receives the user's email and password. This entails creating a special User ID (UID) just for that user. Within the Firebase Authentication system, the UID is essential for locating and managing the user's authentication status and information.

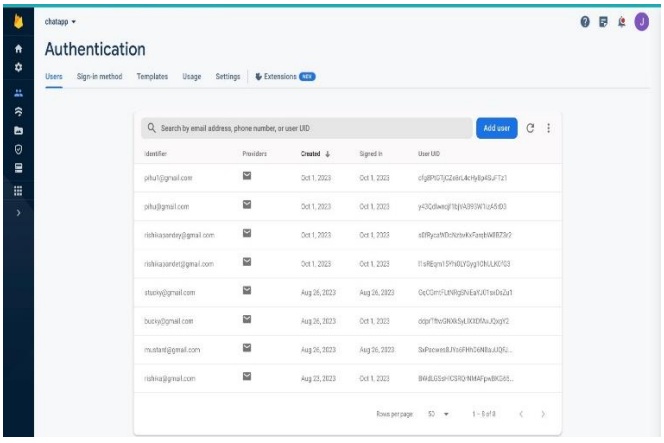

Fig: Screenshot of Firebase Authentication

A created id in the Firebase database indicates the date and time of the chat, the user's email address, the roomId where the chat was written, and finally the chat text. This created id is assigned to all users.

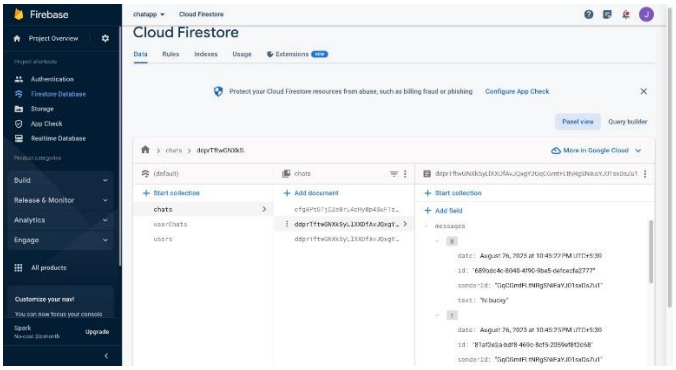

Fig: Screenshot of Chat in Database

VI. CONCLUSION

In conclusion, the Chat Application project is a strong and engaging tool created to promote smooth user interaction. The project includes a clearly defined system architecture with a client-side architecture built on ReactJS and a server-side architecture using Firebase services, as well as seamless integration for real-time updates and seamless user experiences. Because React components are modular and arranged in a hierarchy, a scalable and maintainable frontend is guaranteed. With its real-time database, storage, and authentication features, Firebase functions as a dependable backend. The user interface is simple, and authentication requires a working email and password. Real-time messaging allows for the sending, receiving, and replying of messages, creating a lively and interesting conversation environment.

All things considered, the project successfully integrates frontend and backend technologies to produce a feature-rich, real-time chat application accessible via web browsers across various operating systems.

References:

[1]https://www.lifewire.com/difference-between-chat-and-instant-messaging-3969422.

[2] Shukla, Sanskar, Subhash Chandra Gupta, and Praveen Mishra. "Android-Based Chat Application Using Firebase." 2021 International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2021.

[3] Paolillo, John C., and Asta Zelenkauskaite. "Real-time chat." Pragmatics of computer-mediated communication (2013): 109-133.

[4] Jana, Shraboni, Amit Pande, An Chan, and Prasant Mohapatra. "Mobile video chat: issues and challenges." IEEE Communications Magazine 51, no. 6 (2013): 144-151.

[5] Fedosejev, Artemij. React. js essentials. Packt Publishing Ltd, 2015.

[6] Vipul, A. M., and Prathamesh Sonpatki. ReactJS by Example-Building Modern Web Applications with React. Packt Publishing Ltd, 2016.

[7] Aggarwal, Sanchit. "Modern web-development using reactjs." International Journal of Recent Research Aspects 5.1 (2018): 133-137.

[8]. Maratkar, Pratik Sharad, and Pratibha Adkar. "React JS-An Emerging Frontend JavaScript Library." Iconic Research And Engineering Journals 4.12 (2021): 99-102.

[9]. Rawat, Prateek, and Archana N. Mahajan. "ReactJS: A modern web development framework." International Journal of Innovative Science and Research Technology 5, no. 11 (2020): 698-702.

[10]. Moroney, Laurence, and Laurence Moroney. "The firebase realtime database." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 51-71.

[11]. Moroney, Laurence, and Laurence Moroney. "Using authentication in firebase." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 25-50.

[12]. Moroney, Laurence, and Laurence Moroney. "Using Firebase Hosting." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 93-106.

[13]. Moroney, Laurence, and Laurence Moroney. "Firebase cloud messaging." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 163-188.

[15]. Vamsi Krapa, S.Prayla Shyry, M.Rahul Sai Krishna "WhatsApp Encryption- A Research " International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-2S3, July 2019

[16]. Rösler, Paul, Christian Mainka, and Jörg Schwenk. "More is less: On the end-to-end security of group chats in signal, whatsapp, and threema." In 2018 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 415-429. IEEE, 2018.

[17]. Agrawal, Miss Rachna, Mr Rishabh Rathore, Mr Shashank Jain, Mr SatyamYadav, Mr Vimaljeet Singh, Mr Yash Mishra, and Narendra Pal Singh. "CASE STUDY ON WHATSAPP END TO END ENCRYPTION."

[18]. Henriyan, Diotra, Devie Pratama Subiyanti, Rizki Fauzian, Dian Anggraini, M. Vicky Ghani Aziz, and Ary Setijadi Prihatmanto. "Design and implementation of web based real time chat interfacing server." In 2016 6th International Conference on System Engineering and Technology (ICSET), pp. 83-87. IEEE, 2016.

[19]. GÜLCÜOĞLU, Ekrem, Ahmet Berk USTUN, and Neşet Seyhan. "Comparison of Flutter and React Native Platforms." Journal of Internet Applications and Management 12, no. 2 (2021): 129-143.

[20]. Wieruch, Robin. The road to react: Your journey to master plain yet pragmatic react. js. Robin Wieruch, 2017.

[21]. Graham, Ian S. The HTML sourcebook. John Wiley & Sons, Inc., 1995.

[22] P. K. Kushwaha and M. Kumaresan, "Machine learning algorithm in healthcare system: A Review," 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, 2021, pp. 478-481, doi: 10.1109/ICTAI53825.2021.9673220.

[23] P. K. Kushwaha, B. P. Lohani and D. Singh, "Review on information security, laws and ethical issues with online financial system," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Greater

Noida, India, 2016, pp. 49-53, doi: 10.1109/ICICCS.2016.7542350.

[24]. G. Gulati, B. P. Lohani and P. K. Kushwaha, "A Novel Application Of IoT In Empowering Women Safety Using GPS Tracking Module," 2020 Research, Innovation, Knowledge Management and Technology Application for Business Sustainability (INBUSH), Greater Noida, India, 2020, pp. 131-137, doi: 10.1109/INBUSH46973.2020.9392193.

[25]. D. Pareta, I. N. Verma, B. P. Lohani, P. K. Kushwaha and V. Bibhu, "IoT Enabled Smart and Efficient Musical Water Fountain," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 2022, pp. 369-373, doi: 10.1109/ICIPTM54933.2022.9754129.

[26]. B. P. Lohani, M. Trivedi, R. J. Singh, V. Bibhu, S. Ranjan and P. K. Kushwaha, "Machine Learning Based Model for Prediction of Loan Approval," 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2022, pp. 465-470, doi: 10.1109/ICIEM54221.2022.9853160.

[27]. A. Kumar, B. P. Lohani and P. K. Kushwaha, "Robust Secured Framework for Online Business Transactions over Public Network," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 555-560, doi: 10.1109/ICIEM51511.2021.9445380.

[28]. P. K. Kushwaha and B. P. Lohani, "A review of security of the cloud computing over business with implementation," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Greater Noida, India, 2016, pp. 192-198, doi: 10.1109/ICICCS.2016.7542342.

[29]. M. Chandra, P. K. Kushwaha and S. Saxena, "Modified Fractal Carpets," 2011 International Conference on Computational Intelligence and Communication Networks, Gwalior, India, 2011, pp. 537-540, doi: 10.1109/CICN.2011.115.

[30].P. K. Kushwaha, R. Kohli and D. Singh, "Secret key watermarking in WAV audio file in perceptual domain," 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), Greater Noida, India, 2015, pp. 629-634, doi: 10.1109/ABLAZE.2015.7154940.

[31]. Ranjan, Ankur A. et al. "An Approach for Netflix Recommendation System using Singular Value Decomposition." Journal of Computer and Mathematical Sciences (2019)

[32]. Makkar, Bhavya et al. "Map Reduce concept-based Sentiment Analysis Approach." International Journal of Computer Sciences and Engineering (2019)

[33]. Bhatia, Ayush & Bibhu, Vimal & Lohani, Bhanu & Kushwaha, Pradeep. (2020). An Application Framework for Quantum Computing using Artificial intelligence Techniques. 264-269. 10.1109/INBUSH46973.2020.9392164.

[34]. A. Kumar, B. P. Lohani and P. K. Kushwaha, "Black Hole Attack in Mobile Ad Hoc Network and its Avoidance," 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM), Noida, India, 2021, pp. 103-107, doi: 10.1109/ICIPTM52218.2021.9388366.

[35]. Srivastav, A.V., Lohani, B.P., Kushwaha, P.K., Tyagi, S. (2021). Dual-Layer Security and Access System to Prevent the Spread of COVID-19. In: Prateek, M., Singh, T.P., Choudhury, T., Pandey, H.M., Gia Nhu, N. (eds) Proceedings of International Conference on Machine Intelligence and Data Science Applications. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-33-4087-9_28

[36]. A. Khuran, B. P. Lohani, V. Bibhu and P. K. Kushwaha, "An AI Integrated Face Detection System for Biometric Attendance Management," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 29-33, doi: 10.1109/ICIEM51511.2021.9445295.

[37]. S. Salagrama, B. P. Lohani and P. K. Kushwaha, "An Analytical Survey of User Privacy on Social Media Platform," 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, 2021, pp. 173-176, doi: 10.1109/ICTAI53825.2021.9673402.

[38]. S. Singh, D. Chaudhary, A. D. Gupta, B. Prakash Lohani, P. K. Kushwaha and V. Bibhu, "Artificial Intelligence, Cognitive Robotics and Nature of Consciousness," 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2022, pp. 447-454, doi: 10.1109/ICIEM54221.2022.9853081.

[39]. S. Suman, P. Kaushik, S. S. N. Challapalli, B. P. Lohani, P. Kushwaha and A. D. Gupta, "Commodity Price Prediction for making informed Decisions while trading using Long Short-Term Memory (LSTM) Algorithm," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp. 406-411, doi: 10.1109/IC3I56241.2022.10072626.

[40] P. William, Y. V. U. Kiran, A. Rana, D. Gangodkar, I. Khan and K. Ashutosh, "Design and Implementation of IoT based Framework for Air Quality Sensing and Monitoring," 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2022, pp. 197-200, doi: 10.1109/ICTACS56270.2022.9988646.

[41] Mridul Bhardwaj and Ajay Rana. 2015. Impact of Size and Productivity on Testing and Rework Efforts for Web-based Development Projects. SIGSOFT Softw. Eng. Notes 40, 2 (March 2015), 1–4. https://doi.org/10.1145/2735399.2735404

[42] Bhardwaj, Mridul, and Rana Ajay. "Estimation of testing and rework efforts for software development projects." Asian Journal of Computer Science and Information Technology 5.5 (2015): 33-37.

[43] Dubey, Gaurav, Ajay Rana, and Jayanthi Ranjan. "A research study of sentiment analysis and various techniques of sentiment classification." International Journal of Data Analysis Techniques and Strategies 8.2 (2016): 122-142.

[44] R. Sharma, M. Mogha, S. Tanwar and A. Rana, "Emerging Part of Industry 4.0: The Digital and Physical Technology," 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART), Moradabad, India, 2020, pp. 149-154, doi: 10.1109/SMART50582.2020.9337064.

[45] Dubey, Sanjay Kumar, and Ajay Rana. "Assessment of usability metrics for object-oriented software system." ACM SIGSOFT Software Engineering Notes 35.6 (2010): 1-4.

[46] Singh, Archana, Jyoti Agarwal, and Ajay Rana. "Performance Measure of Similis and FPGrowth Algo rithm." International Journal of Computer Applications 62.6 (2013): 25-31.

[47] Tyagi, Neha, Ajay Rana, and Vineet Kansal. "Load distribution challenges with virtual computing." Intelligent Computing in Engineering: Select Proceedings of RICE 2019. Springer Singapore, 2020.

[48] Singh, Jaya, and Ajay Rana. "Exploring the big data spectrum." International Journal of Emerging Technology and Advanced Engineering 73 (2013).

[49] N. M., P. Chawla and A. Rana, "A Practitioner's Approach to Assess the WCAG 2.0 Website Accessibility Challenges," 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 958-966, doi: 10.1109/AICAI.2019.8701320.

[50] Tyagi, N., Rana, A., Awasthi, S., & Tyagi, L. K. (2022). Data Science: Concern for Credit Card Scam with Artificial Intelligence. In Cyber Security in Intelligent Computing and Communications (pp. 115-128). Singapore: Springer Singapore.

[51] Jain, Piyush, Sanjay Kumar Dubey, and Ajay Rana. "Software usability evaluation method." International Journal of Advanced Research in Computer Engineering & Technology 1.2 (2012): 28-33.

[52] Pal, S. K., et al. "Hanging suicides in himachal pradesh: an analysis of forensic cases." Int J Forensic Sci Pathol 4.11 (2016): 297-304.

[53] Rana, A., and S. Manhas. "Significance of diatoms in diagnosis of drowning deaths: a review." Journal of Forensic & Genetic Sciences 5 (2018): 77-81.

[54] Bansal, Sangeeta, and Dr Ajay Rana. "Transitioning from relational databases to big data." International Journal of Advanced Research in Computer Science and Software Engineering 4.1 (2014): 394-400.