RESEARCH ARTICLE OPEN ACCESS

Analytical description of web based conversation application, XSPARX

Vidyasagar Pandey
(Department of CS, Mangalmay Institute of Engineering and Technology, GB Nagar, India
(Email: vidyasagarpandey43@gmail.com)
Krati Saxena
(Department of CS, Mangalmay Institute of Engineering and Technology, GB Nagar, India
(Email:saxenakrati01@gmail.com)
Shabes
(Department of CS, Mangalmay Institute of Engineering and Technology, GB Nagar, India
(Email:shabeskhan7171@gmail.com)
Dr. Bhaskar Gupta
Dean(Research & Innovation),Mangalmay Institute of Engineering & Technology,GB Nagar,India
(Email:bhaskar.gupta@mietengineering.org)
Mr. Banarsi Lal Prajapati
Assistant Professor,Department of CS, Mangalmay Institute of Engineering and Technology, GB Nagar, India
(Email:banarsi.lal@mietengineering.org)

Abstract:

In the digital age, communication has evolved significantly, and messaging applications have become an essential part of our daily lives. The ability to stay connected in real-time has revolutionized how we interact socially and professionally. While there are several established messaging platforms in the market, the increasing demand for personalized, secure, and efficient communication systems creates an opportunity to build new, innovative solutions.

This project, Xspark, is a web based conversation application designed to meet the basic communication needs of users while offering additional features that enhance the messaging experience. Xspark allows users to create a personalized profile with their name and profile picture, providing a more interactive and engaging experience. Users can log in securely with their credentials, ensuring a seamless and protected entry into the app. The project aims to address the growing need for a secure and efficient communication platform, offering foundational features while laying the groundwork for future scalability and enhancement. This application provides an academic opportunity to apply theoretical concepts in database design, front-end development, and server-side programming, culminating in a fully functional, real-time chat system.

***Keywords* —profile management, group chats, media sharing, interactive conversations.**

I. INTRODUCTION

XSparx is a dynamic and interactive chatting application designed to streamline communication through real-time messaging, a user-friendly interface, and robust backend support. Built using the MERN stack.

Despite the abundance of messaging applications available in the market, such as WhatsApp, Telegram, and Signal, Xspark aims to carve its own niche by offering a simplified yet feature-rich user experience. While these platforms dominate the market, they often come with complex interfaces or lack specific customization options. Xspark strives to combine the most essential features of communication with the flexibility to evolve, allowing future updates to introduce more personalized and unique functionalities. This adaptability ensures that Xspark can stand out in the competitive landscape, meeting users' evolving needs while maintaining ease of use and functionality.

XSparx is a dynamic and interactive chatting application designed to streamline communication through real-time messaging, user-friendly interface, and robust backend support. Developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js), XSparx offers essential features like user authentication, private and group chats, profile customization, and media sharing. The app ensures an engaging user experience by providing real-time notifications, online/offline status indicators, and group management capabilities.

1.1 Features:

User Authentication: Secure login and registration for data protection.

Private and Group Chats: Enables one-on-one and group conversations.

Profile Customization: Allows users to personalize profiles with pictures and details.

Media Sharing: Supports sharing images, videos, and documents.

Real-Time Notifications: Alerts for new messages and updates.

Online/Offline Indicators: Displays user activity status.

Group Management: Enables creation, deletion, and management of group chats.

1.2 Objective:

The primary objective of this project is to develop a fully functional chatting application, Xspark, that meets essential user communication needs. The specific objectives of the project are as follows:

Design a user-friendly interface that allows seamless user registration, profile management, and interaction.

Implement core features including one-on-one chatting, group creation, media sharing (images, videos, documents), and the display of online/offline status.

Integrate secure login mechanisms to ensure user privacy and data protection.

Test the app for functionality, usability, and performance to ensure a smooth and reliable user experience.

Deploy the app and demonstrate its scalability, allowing for future updates and feature enhancements.

Evaluate the app's effectiveness through user feedback and performance testing, identifying areas for future improvements.

AI. Technologies Included

Frontend: React.js for building a responsive and interactive user interface.
Backend: Nodejs and Express.js for server side logics and APIs.
Database: MongoDB for secure and scalable data storage.
Real-time Communication: Socket.IO for realtime messaging and notifications.
Version control: Github for source code management and collaboration.
Deployment tools: Github actions for continuous integration and deployment.

2.1 Hardware requirements:

i. **Processor:** Intel Core i5 or higher

ii. **RAM:** 8 GB or higher

iii. **Storage**: Minimum 20 GB free space

iv. **Internet Connection:** Stable network for real-time communication

2.2 Software requirements:

i. **Operating System**: Windows 10 / Linux / macOS
ii. **IDE:** Visual Studio Code or equivalent
iii. Node.js (LTS version)

iv. MongoDB (Community Server)
v. **Browser:** Google Chrome / Mozilla Firefox.

III.LITERATURE REVIEW

Existing research on messaging apps emphasizes the importance of user-friendly interfaces, real-time communication, and data security. Popular apps like WhatsApp and Telegram dominate due to their seamless messaging and media-sharing capabilities, which have influenced the design of Xspark.

In Xspark, significant progress has been made on the front-end design and database setup, following best practices for UI simplicity and efficient data organization. The database structure ensures scalability and fast data retrieval, inspired by established patterns in messaging apps.

IV. WORKFLOW

User Authentication:

Users can register with their credentials. Login credentials are verified against the database. Successful authentication grants access to the application.

Profile Management:

Users can set up their profiles with a photo, bio, and username. Profiles are stored and retrieved from the database.

Chat Functionality:

Users can engage in private or group chats. Messages are sent and received in real-time using WebSockets. Online/offline status is displayed dynamically.

Group Management:

Group admins can create groups, add/remove members, and manage group settings. Group chats allow multiple users to interact simultaneously.

Media Sharing:

Users can send images, videos, and documents. Uploaded media is stored in a secure backend and linked to the respective chats.

Real-Time Updates:

Online/offline indicators and typing notifications are implemented using socket connections. Chats are updated instantly without refreshing the page.

Database Integration:

MongoDB stores user data, chat history, and media links. Queries are optimized for efficient data retrieval.

Backend Logic:

Node.js and Express.js handle API requests, authentication, and data processing. Middleware ensures secure and seamless communication between the client and server.

Frontend Interface:

React.js creates a dynamic and responsive user interface. Users can navigate through the app with ease, accessing chats, profiles, and settings.

Error Handling:

Failed authentication or server issues display appropriate error messages. Users are prompted to retry actions if needed.

V. IMPLEMENTATION

1. Registration:-Registration is the first step to enter in this chat application. It is a process that requires the user to provide confidential details including their name, email address, and phone number (Sahu, A. K., & Kumar, A., 2021). But in our chat application we will only use the username for user signup. This process can be completed in two ways:

1.1 Enter your user id. -User will enter their user id on login box.

1.2 Choose password: -After entering user id user need to choose their password for authenticating the app in future.

2. Login to Your Account: -A login is a system procedure that allows a user to input their username and password to access a specific account. Before login user must ensure that they have already been registered their self by signing in the app.

2.1 Enter your user id: - In order to login, user will have to enter their registered usernames.
2.2 Enter your password: -After entering username user will enter their password in password box.
2.3 If valid user moves step 3: - If the user's credential match with the database then it will be redirected to chatting page.
2.4 If not a valid user go-to step 1: -In case of not being matched in database, user will have to repeat the step 1 from initial.
3. Find Friends to Start Chatting: -After successful login the user will redirect to chat page where they can see the already registered or available users on the app.
4. Chatting: -The user can initiate its chatting with another available user when they accept their request. User need to type their message in message box. After typing the message user hit the enter button by keyboard or send button by mouse the message will be delivered to the selected user. User can easily chat with friends and check their messages too.
5. Groups: -The user can also create group to chat with more than one user at a time. The message will be delivered to ever member of group with more than one people at a same time.
6. Logout: - To leave or terminate the chatting user can logout from their account. After logout user will be prompt to the login page. The first step in using the chat system is to login. The login is linked to a condition; if the user has not yet registered, he must do so before he may login. They can proceed to the find friend, send message, initiate chat, and logout functions after successfully logging in.

VI. FUTURE SCOPE

XSparx will provide an efficient and secure communication platform, ensuring a smooth user experience with high scalability. Its modular design and feature-rich architecture position it as a competitive tool in the market, with potential applications in business collaboration, education, and social networking.

i. Integration of Voice and Video Calls.

ii. AI-powered Chatbots for automation.

iii. Enhanced Encryption for secure messaging.

iv. Support for Themes and Customization.

VII. CONCLUSIONS

The XSparx project highlights the application of theoretical and practical knowledge in building modern software systems. By utilizing the MERN stack and Agile methodology, the team successfully created a real-time chat application that balances functionality, security, and performance. With scalability at its core, XSparx is ready to meet evolving user demands and integrate advanced features in future iterations.

There is always scope and potential for improvement in every application. Our aim is only engaging with text-based exchanges for now. Other chat software provides services comparable to our product, however they are complicated to use it and have complex layouts. In both human relationships and human-computer interactions, making a strong first impression is critical. The project's goal is to create a chat service Web application with a high-quality user engagement. We are certain that by integrating these services, we will be able to make the application more successful in the future, based on the knowledge gained while developing this application.

□ File Transfer

□ Video Message

□ Audio Call

□ Video Call

□ Group Call

VII. REFERENCES

[1]Goel, A. K., Gupta, S., Singh, C. K., & Agrawal, K. K. (2022). Web-ChatLine: An Innovative Chatting Platform. Materials Proceedings, 10(1), 6.

[2]Gackenheimer, C., & Gackenheimer, C. (2015). Introducing flux: An application architecture for react. Introduction to React, 87-106.

[3]Gupta, K., Btech, C. S. E., Srivastavaa, N., & Goswami, V. (2021). CZAT–A Web Application Based RealTime Chat App.

[4]Shekhawat, A. (2019). WEB BASED CHAT APPLICATION (Doctoral dissertation, GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI).

[5]Sheiko, D.2017). Cross-platform Desktop Application Development: Electron, Node, NW. js, and React. Packt Publishing Ltd.

[6]Tharanidharan, S. K., Yaswanth, P. S., Sastry, M. S. V. C. M., & Sindhwani, M. (2022). Real Time Web Based Multilingual Chat Application. In Futuristic Sustainable Energy and Technology (pp. 33-42). CRC Press

[7]Kumar, T. S., Reddy, V., DL, S., & Rananavare, L. (2021). INTERNET CHAT APPLICATION. International Journal of Advanced Research in Computer Science, 12.

[8]Verma,D. (2022). A comparison of web framework efficiency: performance and network analysis of modern web frameworks.

[9]Biswas, N., & Biswas, N. (2021). Creating a Video Chat Site. Foundation Gatsby Projects: Create Four Real Production Websites with Gatsby, 349-423.