Date of Submission: 12th May,2023

Realtime Chatting Application – FAST CHATAPP

Software Engineering Project

Group Members:

- *Muhammad Zunique (20K-0145)*
- *Kanwar Muzammil (20K-0469)*
- *Awwab Sabir (20K-1615)*
- *Sanan Baig (20K-0165)*

Course Instructor: Miss Hajra

Index:

1.Introduction:

This project report provides an in-depth overview of the development of a real-time chat application that incorporates common features, along with user authentication using email and limited time tokens. The chat application enables seamless communication between users, with additional security measures and the ability to store and share pictures using Cloudinary. The aim of this project is to provide a reliable and user-friendly platform for real-time messaging.

2.Project Overview

The real-time chat application encompasses the following key features:

- User registration and account creation: Users can register for an account using their email address and create a unique username and password.
- Real-time messaging: Users can exchange messages with other users in real-time, enabling instant communication.
- Creation and management of chat rooms: Users have the ability to create chat rooms, invite other users to join, and manage the rooms.
- User authentication using email: The application implements email-based authentication to verify user identities during the registration process.
- Limited time tokens: The application generates limited time tokens for enhanced security, ensuring that user sessions remain secure and active for a specified period.
- User profile management: Users can update their profile information and customize their chat preferences.
- Picture storage using Cloudinary: Cloudinary, a cloud-based media management platform, is utilized to store and manage pictures shared within the chat application.

3. Technologies Used:

The real-time chat application leverages the following technologies:

- Development Stack: **MERN**
- Authentication: Integration with email providers and token-based authentication mechanisms for user verification and session management.
- Cloudinary: A cloud-based media management platform utilized for storing and serving pictures shared within the chat application.

4. System Architecture:

The chat application follows a client-server architecture. The client-side application communicates with the server through a Express API connection, facilitating real-time messaging and interaction. The server is responsible for handling user requests, authenticating users, managing chat rooms, and storing user data and chat history in the MongoDB database. Cloudinary is integrated to manage picture storage and retrieval.

5. Functional Requirements:

The functional requirements of the chat application are as follows:

- User Registration: Users should be able to create an account using their email address, unique username, and password.
- User Authentication: Users must verify their email addresses during the registration process and authenticate themselves using limited time tokens.
- Real-Time Messaging: Users can send and receive messages in real time, enabling instant communication.
- Chat Room Creation and Management: Users have the ability to create chat rooms, invite other users, and manage the rooms.
- User Profile Management: Users can update their profile information, including their display name and profile picture.
- Notifications: Users receive notifications for new messages.

6. Non-Functional Requirements:

The non-functional requirements of the chat application include:

- Performance: The application should be highly responsive and provide real-time messaging capabilities with minimal latency.
- Security: User data, including email addresses and passwords, should be securely stored and transmitted. Token-based authentication ensures session security.
- Scalability: The system should be designed to handle a growing user base and an increasing number of chat rooms and messages.
- User-Friendly Interface: The application should have an intuitive and user-friendly interface, allowing users to navigate and interact with ease.
- Reliability: The chat application should be highly reliable, ensuring that messages are delivered accurately and consistently.
- Data Persistence: User data, chat history shared within the application should be persisted and accessible even after system restarts or failures.

7. Implementation Plan:

The development of the chat application can be divided into the following stages:

1. Requirement gathering and analysis.
2. Designing the application architecture and user interface.
3. Implementing user registration and account creation.
4. Developing real-time messaging functionality using Socket.IO or similar technology.
5. Integrating email authentication and limited time tokens for user verification.
6. Creating chat room creation and management features.
7. Implementing user profile management capabilities.
8. Integrating Cloudinary for picture storage and retrieval.
9. Implementing notifications for new messages and chat room activities.
10. Conducting thorough testing and debugging.
11. Deploying the application to a production environment, utilizing MongoDB with MongoCloud and Cloudinary.
12. Performing user acceptance testing and gathering feedback for further improvements.

8. Conclusion:

The development of a real-time chat application with user authentication using email and limited time tokens provides a secure and seamless communication platform for users. By incorporating the discussed features and following the proposed implementation plan, the chat application aims to deliver a reliable, scalable, and user-friendly experience. With the utilization of MongoDB with MongoCloud for data storage and Cloudinary for picture management, the application ensures efficient handling of user data and media content.

9. Gantt Chart:

Tasks	Start Date	End Date	Duration	Dependencies	Status
1. Develop user interface	15 March,2023	25March,2023	**11 days**	-	Done
2. Implement user authentication	27 March,2023	3 April,2023	**8 days**	-	Done
3. User Search Modal and Add Modal	9 April,2023	11 April,2023	**3 days**	1 and 2	Done
4. Build Single chat message Modal	11 April,2023	17 April ,2023	**7 days**	3	Done
5. Build Group chat message Modal	19 April,2023	27 April,2023	**9 days**	3	Done
6. Conduct user testing	28 April,2023	30 April ,2023	**3 days**	2,3,4,5	Done
7. Address bugs and Issues	1 May,2023	2 May,2023	**2 days**	6	Done

Total Project Duration: 43 days (including off days)