ऋते ज्ञानान्न मुक्तिः

GLA UNIVERSITY
MATHURA
Established vide U.P. Act 21 of 2010.

SYNOPSIS

ON

Chat Application

Submitted By:

Abhishek Kumar - 2115000021

Advik Saxena - 2115000084

Garvit Dewan -2115700002

Harshit Sharma - 2115700003

Submitted To:

Mr. Ankit Arora

Assistant Professor

Department of Computer Engineering and Application

Title of the Project:

BuzzTalk: Your real-time chat solution

Objective:

Instant Messaging: Chat applications allow users to send and receive text messages.

Remote Collaboration: They provide a platform for remote teams to collaborate, discuss work-related matters, and share files or documents, thus helping organizations enhance productivity and teamwork.

Personal Communication: Chat apps are used for personal communication, enabling friends and family to stay in touch, share updates, and maintain social connections.

Group Communication: Chat applications often support group chats or channels, which can be used for various purposes, including project coordination, interest-based discussions, or social groups.

Notification and Alerts: They can be used for sending important notifications and alerts to users, such as news updates, weather alerts, or event reminders.

Scope:

Real-time Messaging: Chat applications typically provide real-time messaging capabilities, allowing users to send and receive text messages.

User Authentication: Chat applications require user authentication to ensure that messages are sent and received securely. This may involve registration, login, and password reset functionality.

Boundaries and Limitations:

Scalability: Building a chat application that can handle a large number of users and messages concurrently can be complex. Scalability is a significant challenge.

Server Infrastructure: Developing and maintaining the server infrastructure required for real-time communication can be resource-intensive.

Methodology:

Programming Languages: JavaScript for web-based chat applications, Node.js for server-side development.

Development Frameworks and Libraries: React for building the user interfaces of web-based chat applications.

Real-Time Communication: Web Socket, Socket.IO

Databases: MongoDB for storing user data and chat messages.

Proposed System:

The proposed system is a real-time chat application designed for both personal and business use. Its core idea is to provide a user-friendly, secure, and feature-rich platform for real-time communication and collaboration. The application will support one-on-one and group chats, multimedia sharing, user authentication, and other essential features commonly found in chat applications.

Features:

User Registration and Authentication, User Profiles, Real-Time Messaging, Group Chats, Notifications, Night Mode

Implementation Plan:

- Project Planning Phase (1-4 days)
- Design Phase (1-5 days)
- Development Phase (1-2 weeks)
- Testing and Quality Assurance (1-2 weeks)
- Deployment and Launch (1-4 days)

Team Members:

- Abhishek Kumar, Harshit Sharma -UI/UX designer
- Garvit Dewan , Advik Saxena-Frontend developer
- Abhishek , Advik Saxena=Backend developer

Resources Required:

- Integrated Development Environment (IDE): A suitable IDE for writing, testing, and debugging code. Common choices include Visual Studio Code
- Version Control System: Git is the most commonly used version control system, and you'll need a platform like GitHub for code collaboration and version control.
- Project Management Tools: Software like Jira, Trello, Asana, or similar tools for project planning, task management, and team collaboration.
- Database Management System: a database management system based on our project's requirements include MySQL, MongoDB.
- Web Development Frameworks: Depending on our chosen technologies, we will need web development frameworks (e.g., React) for the web version.
- Real-Time Communication Libraries: Libraries like Socket.io, WebSocket's, or Firebase Realtime Database for implementing real-time messaging.

References:

- Online Documentation of React, Node.js, and Soket.io etc.
- GitHub Repositories to see how other developers have implemented various feature in their chat application.

Expected Outcomes:

- Working Chat Application
- Scalable infrastructure
- User Engagement
- Documentation
- User Adoption and growth

Project Supervisor:

Mr. Ankit Arora

Conclusion:

A. Core Objective: To create a user-friendly, secure, and feature-rich chat application for personal and business use.

B. Key Features: The application will include user registration, real-time messaging, group chats, notifications, night mode, security and privacy features, and cross-platform availability.

C. Scalability**:** Designing a server infrastructure that can accommodate a growing user base without compromising performance.

D. Quality Assurance: Rigorous testing and ongoing maintenance to ensure a high-quality user experience.

E. User Engagement and Growth: Promoting user adoption, engaging users effectively, and adapting to user feedback for continuous improvement.