Chatflow

A PROJECT REPORT
for
Mini Project-II

Submitted by

Mahima Goyal
(202410116100112)
Krishna
(202410116100103)
Kartikey Dwivedi
(202410116100098)
Kunal Prajapati
(202410116100108)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms Shruti Aggarwal
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad Uttar
Pradesh-201206

March-2025

CERTIFICATE

Certified that **Mahima Goyal (202410116100112), Krishna (202410116100103) and Kartikey Dwivedi (202410116100098)** and **Kunal Prajapati (202410116100108)** has/ have carried out the project work having "**Chatflow**" (**Mini Project-II**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Ms Shruti Aggarwal	**Dr. Akash Rajak**
Assistant Professor	**Dean Department of Computer Applications**
Department of Computer Applications	**KIET Group of Institutions, Ghaziabad KIET**
Group of Institutions, Ghaziabad	

“Chatflow”

Mahima Goyal
Krishna
Kunal Prajapati
Kartikey Dwivedi

ABSTRACT

ChatFlow is a real-time, anonymous chat website designed to facilitate secure and private communication without requiring users to register or share personal details. Built using **Java Spring Boot** for the backend, **JavaScript** for the frontend, and a **relational database (MySQL/PostgreSQL)** for session management, the platform leverages **WebSocket technology** for instant, low-latency messaging.

The project focuses on providing a **seamless and anonymous** chatting experience while ensuring data security through **encryption, session-based storage, and automatic message deletion**. Key features include **instant user pairing, a responsive UI, spam prevention mechanisms, and cross-device compatibility**.

As the project is currently in development, challenges such as **real-time performance optimization, handling concurrent users, and implementing robust security measures** are being addressed. Planned enhancements include **group chat support, end-to-end encryption, a mobile application, and AI-based spam detection**.

ChatFlow aims to be a **scalable, user-friendly, and privacy-centric** communication platform, ensuring an engaging and safe chatting experience. The final product will provide a **secure, real-time messaging solution** that balances anonymity with functionality.

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms Shruti Aggarwal** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Akash Rajak, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions. Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Mahima Goyal
Krishna
Kunal Prajapati
Kartikey Dwivedi

TABLE OF CONTENT

Introduction

- In the digital era, online communication has become an integral part of daily life, but privacy concerns continue to rise. ChatFlow is a real-time anonymous chat website designed to provide secure and private communication without requiring users to disclose personal information. By leveraging WebSocket technology, ChatFlow ensures instant, lag-free messaging, making conversations more interactive and efficient.
- The platform is built using Java Spring Boot for the backend, JavaScript for the frontend, and MySQL/PostgreSQL for database management. WebSockets facilitate real-time bidirectional communication, eliminating delays associated with traditional HTTP requests.
- **Key Highlights of ChatFlow:**
- Anonymous Communication: Users can chat without registering or providing any personal details.
- Real-Time Messaging: WebSocket technology ensures instant message transmission.
- Secure and Private: Encryption and session-based data storage enhance user security.
- Scalability: The system is designed to handle a large number of users efficiently.
- User-Friendly Interface: A simple, intuitive UI for seamless user interaction.
- ChatFlow is an ideal solution for individuals seeking private conversations without compromising their security and anonymity. This project aims to bridge the gap between privacy and convenience in online communication.

Feasibility Study

A feasibility study was conducted to assess the viability of ChatFlow from different perspectives. This study helps determine the practicality of the project and its ability to be successfully developed and implemented.

3.1 Technical Feasibility

ChatFlow is technically feasible due to the availability of robust web development tools and frameworks. The project leverages:

- **Backend:** Java Spring Boot, WebSocket for real-time communication.
- **Frontend:** JavaScript, HTML, CSS, React (optional) for a modern UI.
- **Database:** MongoDB for structured data storage.

3.2 Economic Feasibility

ChatFlow is a cost-effective solution because it is built using **open-source technologies**, minimizing licensing costs. Additional factors include:

- **Minimal hardware requirements:** Standard web servers are sufficient.
- **Low operational costs:** No need for extensive IT infrastructure.

3.3 Operational Feasibility

ChatFlow is designed to be user-friendly and require minimal training. Key operational aspects include:

- **Simple UI:** Users can start chatting without complex navigation.
- **Automatic Pairing:** No manual user selection needed.
- **Session-Based Chat:** Ensures minimal data storage requirements.

3.4 Schedule Feasibility

The project follows a structured **two-month timeline** with clearly defined milestones:

- **Month 1:** Frontend design, backend integration, WebSocket implementation.
- **Month 2:** Security enhancements, testing, optimization, and deployment. With a dedicated team and proper planning, the project is achievable within the stipulated time.

3.5 Social Feasibility

ChatFlow promotes **privacy and anonymous communication**, making it suitable for:

- **Support groups and counseling sessions.**
- **General anonymous social interactions.**

Project Objectives

1 The primary objective of ChatFlow is to develop a secure, real-time, and anonymous chat platform that allows users to communicate without revealing their identity. The system is designed to ensure a seamless chatting experience while maintaining privacy and security.

Key Objectives:

- User Anonymity: Users can chat without registering or providing personal details.
- Real-Time Messaging: WebSocket-based instant message transmission.
- Secure Communication: End-to-end encryption and data protection mechanisms.
- Scalable Architecture: Handle multiple concurrent users efficiently.
- Minimal Data Retention: Session-based chat storage without permanent logs.
- Spam Prevention: Implement CAPTCHA, rate limiting, and AI-based moderation.
- Cross-Platform Compatibility: Ensure a responsive UI for desktops and mobile devices.
- Automatic User Pairing: Connect users randomly for spontaneous conversations.
- Future Scalability: Support for group chats, AI moderation, and additional security enhancements.

By achieving these objectives, ChatFlow aims to provide a privacy-centric chat platform that balances security, usability, and performance.

Hardware and Software Requirements

4.1 Hardware Requirements:

- **Server Requirements:** ○ Minimum 8-core processor (Intel Xeon or AMD Ryzen preferred) ○ 16 GB RAM or higher ○ 500 GB SSD storage
  - High-speed internet connection for WebSocket communication ○ Load balancer for handling multiple connections (optional for scaling)
- **Client Requirements:**
  - Any modern device (PC, laptop, tablet, smartphone) ○ Internet connection
  - Updated web browser (Chrome, Firefox, Edge, Safari)

4.2 Software Requirements

- Operating System: ○ Windows, macOS, or Linux (for development and hosting)
- Backend Development: ○ Java Spring Boot ○ WebSocket API
  - Hibernate (for database management)
- Frontend Development:
  - HTML, CSS, JavaScript (React optional) ○ WebSocket client implementation
- Database Management:
  - MongoDB
- Server & Hosting:
  - Apache/Nginx
  - AWS/DigitalOcean for cloud deployment
- Development Tools:

- IntelliJ IDEA / Eclipse (for Java development) o Visual Studio Code (for frontend development)
- Postman (for API testing)

Timeline for Completion

The development of ChatFlow is structured to be completed within the next **two months**, ensuring steady progress with achievable milestones. The revised timeline for completion is as follows:

Month 1: Frontend Development & Backend Setup

Week 1-2: UI/UX Design and Initial Development •

Develop wireframes and interactive UI prototypes.

- Ensure a responsive, user-friendly design.
- Gather feedback from initial users for improvement.
- Implement the designed UI using HTML, CSS, and JavaScript (React optional).

Week 3-4: WebSocket and Backend Integration

- Develop WebSocket-based backend for real-time messaging.
- Set up Java Spring Boot backend with API development.
- Establish a relational database (MySQL/PostgreSQL) for session management.
- Ensure seamless frontend-backend communication.

Month 2: Testing, Security Enhancements, and Deployment Week 5: Security Enhancements and Performance Optimization

- Implement encryption for secure messaging.
- Apply authentication and spam prevention mechanisms.
- Optimize database queries and improve system performance.

Week 6: System Testing and Debugging

- Conduct load testing for high-traffic scenarios.
- Perform integration testing to ensure smooth user experience.
- Identify and resolve critical performance issues.

Week 7: Deployment and Documentation

- Set up cloud deployment (AWS/DigitalOcean).
- Implement CI/CD pipelines for continuous integration.
- Prepare project documentation, including user guides and API references.

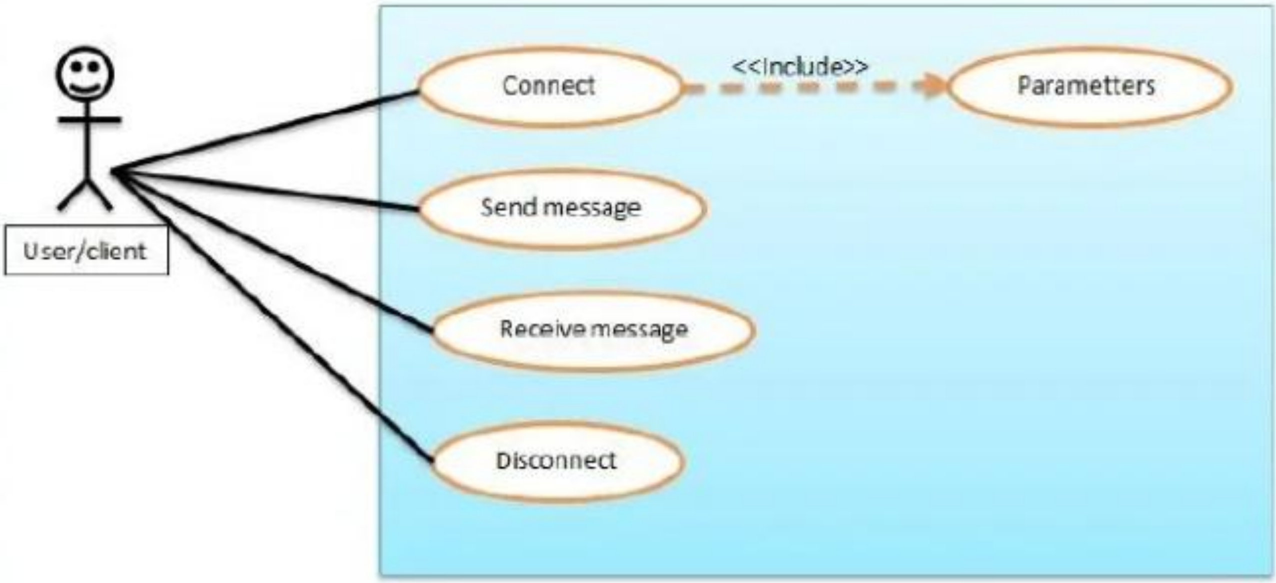
Week 8: Final Optimization and Project Completion

- Conduct final quality assurance checks.

- Optimize the application based on user feedback.
- Prepare for final demonstration and project submission.

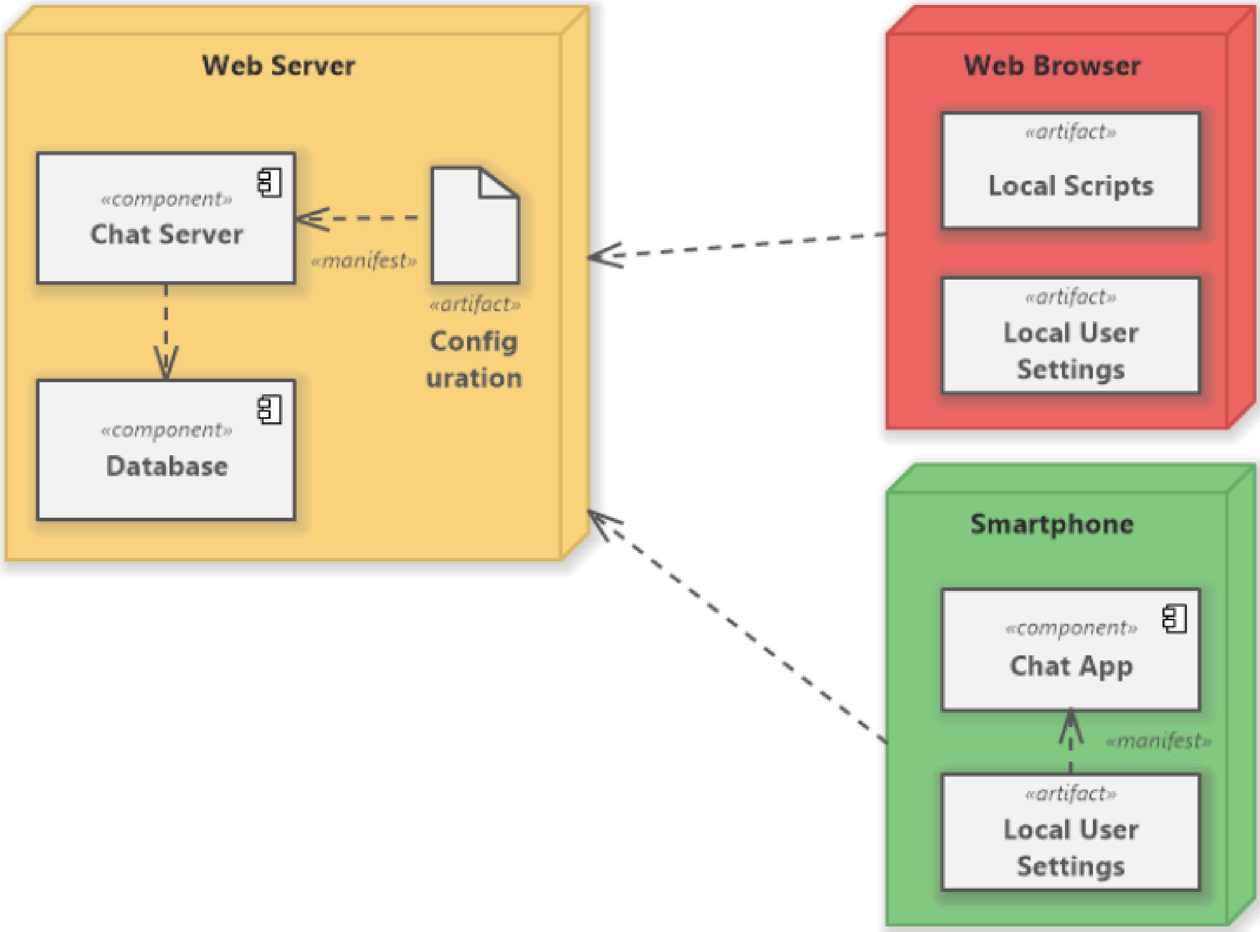
USE CASE DIAGRAM

Use case

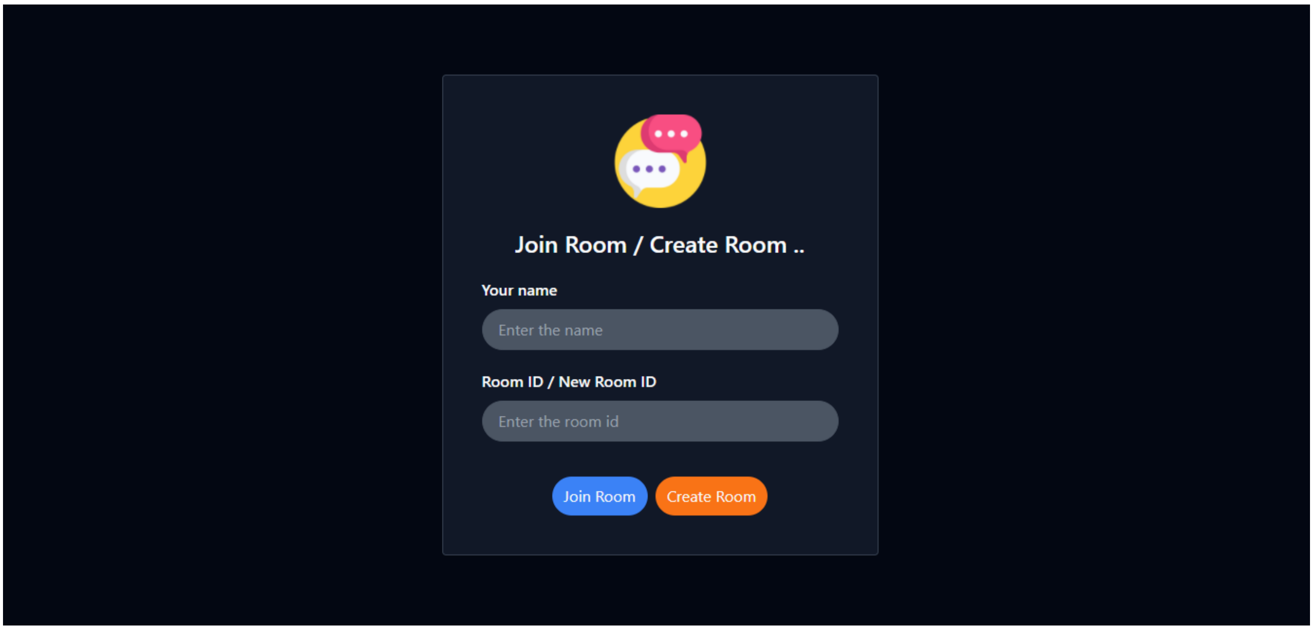


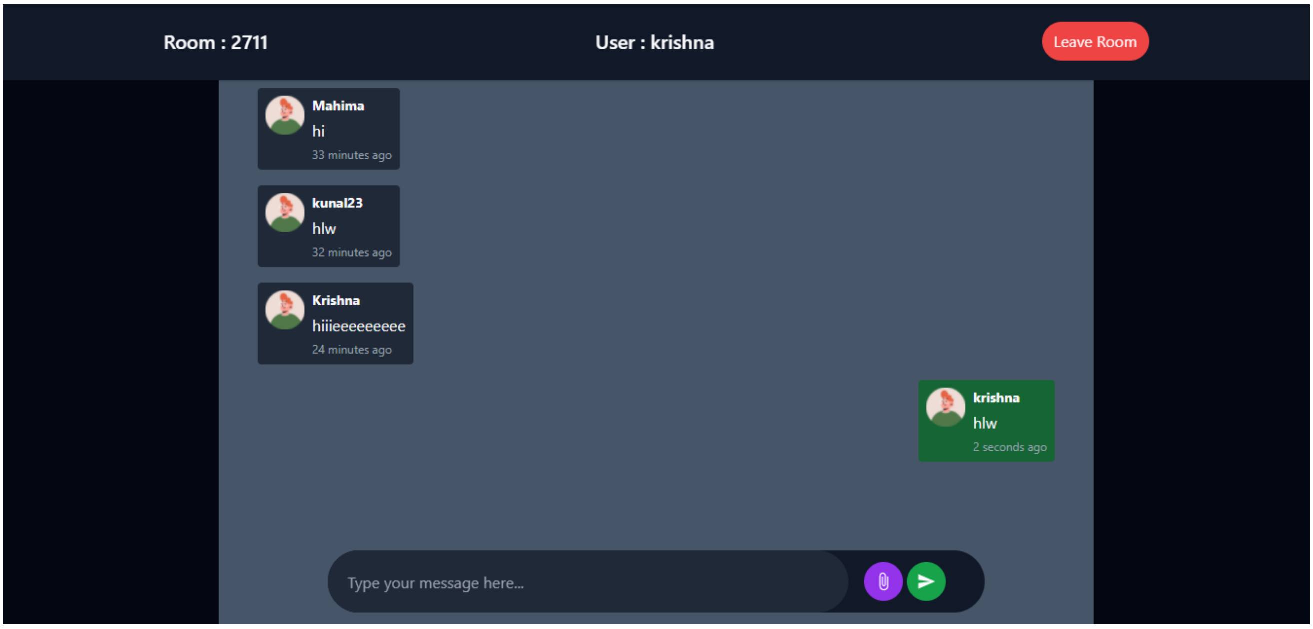
Client chat system

DATA FLOW DIAGRAM



OUTPUT




Room : 2711

User : krishna

Mahima
hi
33 minutes ago

kunal23
hlw
32 minutes ago

Krishna
hiiieeeeeeeee
24 minutes ago

krishna
hlw
2 seconds ago

References:

1. Smith, J., & Lee, K. (2022). *Real-Time Messaging Using WebSocket: A Comparative Study*. International Journal of Computer Science, 45(3), 120-135.
2. Brown, T., & Williams, D. (2021). *Ensuring Privacy in Anonymous Chat Systems*. Cybersecurity Journal, 10(2), 89-102.
3. Johnson, R. (2020). *Building Scalable Backend Systems with Spring Boot*. Software Engineering Journal, 33(4), 56-72.
4. **Spring Boot WebSocket Guide** - Official Spring documentation on implementing WebSockets: https://spring.io/guides/gs/messaging-stompwebsocket/
5. **React WebSocket Integration** - MDN Web Docs: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API 6. **AWS Deployment Best Practices -** https://docs.aws.amazon.com/