(AnnexureB)

INTEGRATED PROJECT REPORT

On

Talk-A-Tive

Submitted in partial fulfillment of the requirement for the Course
Full-Stack Engineering Project (CS 203) of

COMPUTER SCIENCE AND ENGINEERING
B.E. Batch-2022 (VI Semester)

SubmittedBy

Under the Guidance of
Mr. Rahul Sir

Name: Pratham Midha
Roll. No. : 2210992076
Name: Prabhat Kumar
Roll. No. : 2210992048
Name: Prince
Roll. No. : 2210992088
Name: Prince Kumar
Roll. No. :2210992090

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CHITKARA UNIVERSITY
PUNJAB

(Annexure–C)

CERTIFICATE

This is to be certified that the project entitled “Talk-A-Tive” has been submitted for the Bachelor of Computer Science Engineering at Chitkara University, Punjab during the academic semester January 2025-May-2025 is a bonafide piece of project work carried out by “Prince (2210992088), Prince kumar (2210992090),Prabhat kumar (2210992048), Pratham Midha (2210992076)” towards the partial fulfillment for the award of the course FSE Project (CS 203) under the guidance of “Mr. Rahul Sir” and supervision.

Sign. of Project Guide

Dev Insights

Mr. Rahul Sir

(Annexure–D)

CANDIDATE'SDECLARATION

We, **"Prince (2210992088), Prince kumar(2210992090), Prabhat Kumar (2210992048), Pratham Midha(2210992076)",** B.E.-2022 of the Chitkara University, Punjab hereby declare that the Full Stack Engineering Project Report entitled **"Talk-A-Tive"** is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other course.

Sign.of Student1	**Sign.of Student2**	**Sign.of Student3**	**Sign.of Student4**
Prince	Prince kumar	Prabhat Kumar	Pratham Midha
ID No. 2210992088	ID No. 2210992090	ID No. 2210992048	ID No. 2210992076

Place: Rajpura
Date: 3-03-2025

(Annexure-E)

ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior and acts during the course of study.

We express our sincere gratitude to all for providing mean opportunity to undergo FSE Project as the part of the curriculum.

We are thankful to "Mr. Rahul Sir" for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We also extend our sincere appreciation to **Mr. Rahul** Who provided his valuable suggestions and precious time in accomplishing our Talk-A-Tive project report.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we share dour day-to day experience and received lots of suggestions that improve our quality of work.

Prince	Prince kumar	Prabhat Kumar	Pratham Midha
ID No. 2210992088	ID No. 2210992090	ID No. 2210992048	ID No. 2210992076

CHITKARA UNIVERSITY

(Annexure–F)

1. **Abstract/Keywords**
2. **Introduction to the project**
    - **2.1 Background**
    - **2.2 Problem Statement**
3. **Software and Hardware Requirement Specification**
    - **3.1 Methods**
    - **3.2 Programming/Working Environment**
    - **3.3 Requirements to run the application**
4. **Database Analyzing, design and implementation**
5. **Program's Structure Analyzing and GUI Constructing**
6. **Code-Implementation and Database Connections**
7. **System Testing**
8. **Limitations**
9. **Conclusion**
10. **Future Scope**
11. **Bibliography/References**

CHITKARA UNIVERSITY

(Annexure–F)

1. Abstract/Keywords

Talk-A-Tive is a real-time messaging platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), aimed at delivering fast, secure, and engaging communication experiences. It features one-on-one messaging, group chats, and a modern UI that ensures intuitive user interaction. With WebSocket integration for instant message delivery and Firebase Authentication for secure access, Talk-A-Tive offers a reliable and responsive chatting environment. Designed with Tailwind CSS and deployed on Vercel, the application combines sleek design with efficient functionality to create a seamless user experience.

Keywords:
Talk-A-Tive, MERN Stack, Real-Time Messaging, WebSocket, React.js, Node.js, Express.js, MongoDB, Firebase Authentication, Group Chat, Tailwind CSS, Vercel Deployment

2. Introduction to the Project

Traditional communication platforms often suffer from performance bottlenecks, complex user interfaces, or a lack of real-time interaction, making digital conversations less efficient and engaging. **Talk-A-Tive** solves this by offering a clean, real-time messaging platform built on the MERN stack. It provides users with secure authentication, one-on-one and group messaging, and a responsive UI, ensuring smooth communication across all devices. By leveraging technologies like WebSockets, Firebase, and modern frontend frameworks, Talk-A-Tive ensures fast, intuitive, and reliable digital conversations for both casual and professional users.

2.1 Background

With the growing dependence on online communication, the need for scalable, responsive, and real-time messaging platforms has significantly increased. Many traditional chat systems lack modern features such as instant delivery, proper authentication, or group messaging capabilities. Furthermore, complex UIs and delayed responses reduce user satisfaction. **Talk-A-Tive** is developed using the MERN stack, integrating Firebase Authentication, WebSocket-based real-time messaging, and Tailwind CSS to deliver a feature-rich and seamless chatting experience.

2.2 Problem Statement

The major issues in current chat applications include:

- **Lack of Real-Time Communication:** Many platforms suffer from delays in message delivery, disrupting the natural flow of conversation.
- **Poor UI/UX Experience:** Complex and outdated interfaces reduce user engagement and ease of use.
- **Limited Group Chat Features:** Some applications provide only basic messaging functionality without robust group interaction tools.
- **Security Concerns:** Insecure login systems and lack of proper authentication pose a risk to user data and privacy.
- **Scalability Limitations:** Applications not built on modern web technologies may fail to handle large numbers of users or real-time updates effectively.

3. Software and Hardware Requirement Specification

3.1 Methods

Talk-A-Tive adopts the Agile development methodology to ensure continuous iteration, user feedback integration, and adaptive planning throughout the project lifecycle. The application is developed using modern web technologies and real-time communication protocols, focusing on scalability, performance, and a seamless user experience.

CHITKARA UNIVERSITY

(Annexure–F)

3.2 Programming/Working Environment

Talk-A-Tive is developed using the following technologies:

- **Frontend Technologies:** React.js, Tailwind CSS
- **Backend Technologies:** Node.js, Express.js
- **Database Management:** MongoDB (using Mongoose)
- **Authentication:** Firebase Authentication
- **Real-Time Communication:** WebSocket (Socket.IO)
- **Deployment:** Vercel (Frontend), Render/Heroku (Backend if separate)

3.3 Requirements to Run the Application

• Hardware Requirements:

- **Processor:** Intel i5 or higher
- **RAM:** Minimum 4GB (Recommended: 8GB)
- **Storage:** Minimum 20GB of free space
- **Display:** Minimum 1024x768 resolution

• Software Requirements:

- **Operating System:** Windows, Linux, or macOS
- **Development Tools:** Visual Studio Code, Node.js, Git
- **Database:** MongoDB (local or MongoDB Atlas)
- **Browser:** Google Chrome, Firefox, or Microsoft Edge
- **Other:** Internet connection for real-time messaging and Firebase services

4. Database Analyzing, Design, and Implementation

Talk-A-Tive uses **MongoDB** to manage real-time messaging data with scalability and efficiency. The database is designed to support quick retrieval and secure storage of user and message information.

Database Design:

- **Users Collection:** Stores user info (ID, name, email, profile pic, status).
- **Messages Collection:** Stores individual messages (sender, receiver, content, timestamp).
- **Chats Collection:** Manages one-on-one chat metadata.
- **Groups Collection:** Stores group chat details (name, members, image).
- **GroupMessages Collection:** Stores messages specific to group chats.

Normalization:

Data is logically organized following **3NF principles** to reduce redundancy, improve performance, and ensure clean data relationships.

5. Program's Structure Analyzing and GUI Constructing

5.1 High-Level Design:

- **Flowchart & Use Case:** Shows user login, chat initiation, and group chat interactions.
- **Architecture:**
  - **Frontend:** React.js, Tailwind CSS
  - **Backend:** Node.js, Express.js
  - **Database:** MongoDB
  - **Authentication:** Firebase
  - **Real-Time:** WebSocket (Socket.IO)

5.2 GUI Screenshots:

- **Login Page:** Secure sign-in with Firebase.
- **Dashboard:** Lists chats and contacts.

(Annexure–F)

- **Chat Screen:** Real-time one-on-one messaging.
- **Group Chat:** Group creation, messaging, and member display.

6. Code Implementation and Database Connections

- **Frontend Code: Built using React.js and Tailwind CSS for a responsive, user-friendly UI.**
- **Backend Code: Developed with Node.js and Express.js to handle authentication, real-time messaging, and chat features.**
- **Database Connection: MongoDB used for storing user data, chats, and messages, with efficient data management.**

---

7. System Testing

Types of Testing:

- **Unit Testing: Verifies individual components and API endpoints.**
- **Integration Testing: Ensures frontend, backend, and database work together seamlessly.**
- **User Acceptance Testing (UAT): Validates the app's user-friendliness and intuitiveness.**

---

8. Limitations

- **AI Accuracy: Recommendations may not always be perfectly tailored to each user.**
- **API Dependencies: External APIs like Firebase and WebSocket may cause occasional issues.**
- **Internet Dependency: Requires an active internet connection for real-time messaging and updates.**

---

9. Conclusion

- **Innovative Platform: Combines real-time messaging and seamless UI for efficient communication.**
- **User-Friendly & Scalable: Built with React.js, Node.js, and MongoDB for optimal performance.**
- **Challenges Overcome: Real-time communication and secure authentication.**
- **Future Potential: Plans to expand features, including mobile app integration.**

---

10. Future Scope

- **User Engagement: Implement challenges and reward systems.**
- **Personalized Experience: Gender-specific features like period tracking for tailored fitness.**
- **Seamless UI/UX: Improved responsiveness with Tailwind CSS.**
- **Community Features: Leaderboards, social interaction, and progress tracking.**
- **Scalability & Security: Optimized for multiple users with secure data handling.**

---

11. Bibliography/References

- **React.js Documentation: react.dev**
- **Node.js & Express.js Docs: nodejs.org**
- **MongoDB Documentation: mongodb.com/docs**
- **Firebase Authentication: firebase.google.com**
- **Socket.IO: socket.io**