GLA UNIVERSITY

TOPIC : SYNOPSIS ON REALTIME CHAT APP

Submitted by

Akansh Jain
(201500055)
KaustubhYadav
(201500333)
Sourabh Thakur
(201500703)

Submitted to

Mr. Akash Kumar Choudhary
Technical Trainer

DECLARATION

We hereby declare that the project work entitled **"Real Time Chat App"** submitted to the GLA University, is a record of an original work done by us group mates under the guidance of **Mr. Akash Kumar Choudhary**, and this project work is submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Acknowledgement

It gives us a great sense of pleasure to present the synopsis of the project undertaken during B.Tech III Year. This project is going to be an acknowledgement to the inspiration, drive and technical assistance will be contributed to it by many individuals. We owe special debt of gratitude to Mr. Akash Kumar Choudhary, Technical Trainer , for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal and for his constant support and guidance to our work.

His sincerity, thoroughness and perseverance has been a constant source of inspiration for us. We believe that he will shower us with all his extensively experienced ideas and insightful comments at different stages of the project & also taught us about the latest industry-oriented technologies. We also do not like miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and co-operation.

Akansh Jain (201500055)
Kaustaubh Yadav (201500333)
Sourabh Thakur (201500703)

Contents

INTRODUCTION

The emergence of modern technologies has had profound impacts on the education landscape, with online learning now an integral part of the learning process. The main advantages of online learning are flexibility and accessibility. Student access to educators to assist them is no longer restricted to the hours of operation of schools and universities, but can be provided anytime and anywhere. Face-to-face tutoring is a well-established, and effective, instructional method. However, there is a need for more empirical research to be directed toward investigating users' experiences with online tutoring services, their impact on academic confidence (self-efficacy), and achievement scores. The purpose of this project is to develop a back-end application for a chat application and queries using graphical user interface

User can communicate in realtime effortlessly and can manage their records of chat.

ABSTRACT

The purpose of Realtime Chat Application is to provide the user error free , secure , reliable and fast management app . A Real time chat application makes it easy to communicate with people anywhere in the world by sending and receiving messages in real time. This project helps user to keep the records of their for a longer period of time.

Details About the Hardware and the Software

Software and hardware requirements

- A Mac, Linux, or Windows 10 or Windows 11 computer
- An internet connection
- A web browser like Chrome or Microsoft Edge

Frontend and backend

- FRONTEND – JSX, REACT JS, BOOTSTRAP
- BACKEND – NODE JS, EXPRESS JS, MONGO DB

IDEA OF THE PROJECT

The idea of this project is to develop an application where user can interact with each other which makes it user friendly and extensible. It is equipped with a vast array of features that makes it suited for a number of the scenarios

The Main objective of the Project.

The main objective of the RealTime Chat project is to create a platform where peoplecan easily communicate with people anywhere in the world by sending and receiving the messages in real time. The project also provides an easy way for users to sign up and login, and should also allow them to create and manage their own chat rooms. The application will be built using a React front-end that can be used by anyone, anywhere. The project allows users to send and receive messages in real-time and will give a user-friendly interface for users to interact with.

Scope Of the Project

The Realtime chat Application project is a chat application that allows users to communicate with each other in real time. It provides a set of tools for building and managing chat applications, as well as a set of APIs for integrating chat applications with other web applications. The project is open source and available on GitHub. The application has the following features:

- Users can create a profile and add friends.
- Users can send and receive messages
- Users can create and join groups.
- Users can share files and images
- Users can search for other users
- Users can block other users

Working Methodology of the Realtime chat Project

The working methodology of the Realtime Chat Appliaction project is very simple and easy to understand. The project is developed using the MERN stack. The MERN stack is a collection of JavaScript-based technologies used for developing web applications. The MERN stack includes the MongoDB database, the Express.js web application framework, the ReactJS web application framework, and the Node.js runtime environment. The realtime Chat Application project makes use of the Socket.IO library toenable real-time communication between the client and the server.

The project consists of two parts: the front-end and the back-end. The front-end is implemented using ReactJS, and the back-end is implemented using Node.js. The front-end of the chat application is responsible for the user interface and the user experience. It handles all the user input and output, and it communicates with the back-end using the ReactJS $http service. The back-end of the chat application is responsible for the business logic and the data storage. It communicates with the front-end using the Node.js Express.js framework.

Module description

The purpose of this project is to develop a back-end application for real time messaging. It allows multiple users to interact with each other where they can chat.

The project is divided into 3 modules – Sign-In, Sign-up, Dashboard. The roles of the modules are as follows:

- **Sign-up:**

The sign up page allows a user to create an account by their credentials to gain access to application

- **Sign-in:**

The login page allows a user to gain access to an application by entering their username and password. A user navigates to an application and is presented with a login page as a way to gain access to the application. There are two possible results:

  - Authentication is successful and the user is directed to the application landing page.
  - Authentication fails and the user remains on the login page. If authentication fails, the screen should show an informational or error message about the failure.

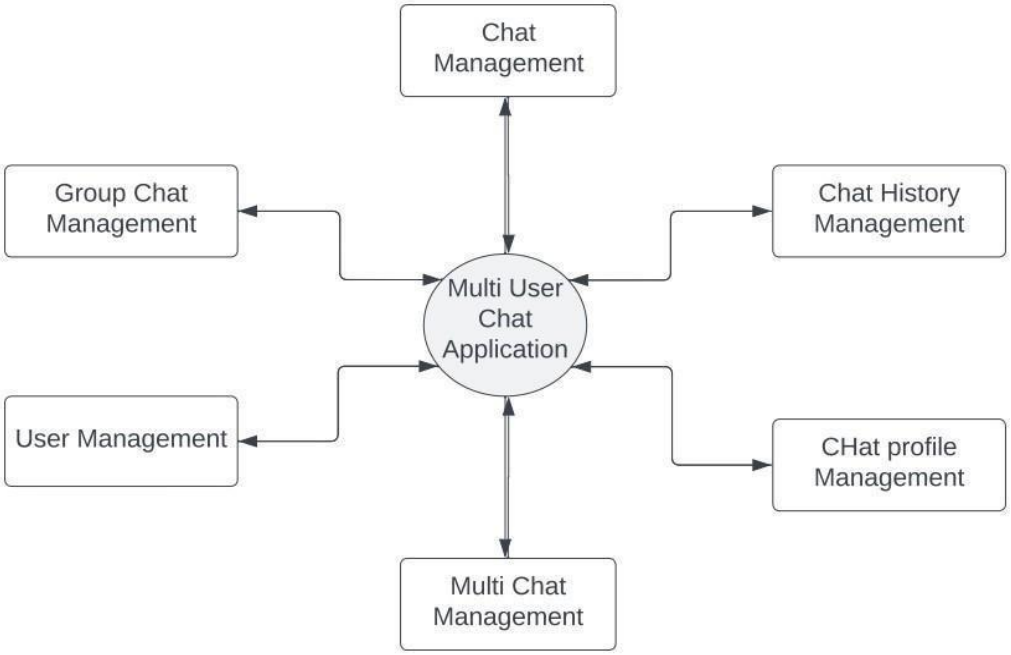
- A user is automatically logged out due to inactivity. In this event, they will be returned to the login page, which will display an informational message explaining what happened. Once the user logs in again, they should be taken back to the page they were previously on before being timed out. Thirty minutes is the suggested duration before a session timeout, but this is subject to change based on your product's security requirements.
- A user has forgotten their username and/or password. A link is available to begin the process to reset this information. Once the user clicks on one of these links, the contents of the login page is replaced with fields specific to recovering their username and/or password. There are a number of different ways the user could recover their password. This pattern does not dictate which methods an application should follow. Some options include:
    - The user could provide their e-mail and be sent a temporary password or a link to reset their password.
    - The user could answer a security question.
    - The user could get a message explaining that they have to contact a specific person

- **Dashboard:**

This is where user can see themselves to the very first interface of the application . Usernames are searchable and they can use them to connect with each other with their respective e mails through which users are logged in .

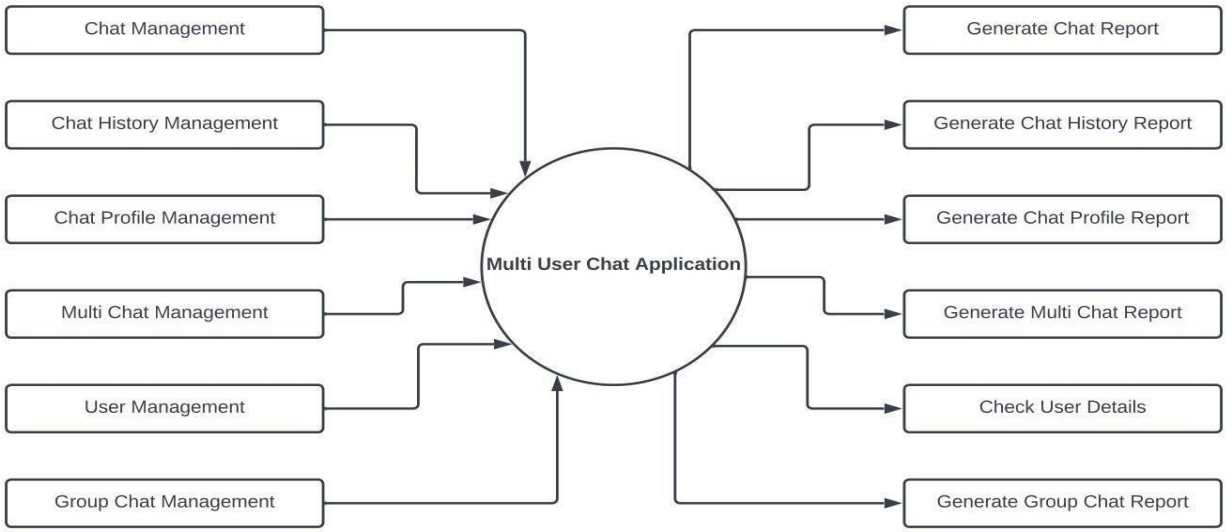
Data Flow Diagrams

0 Level DFD



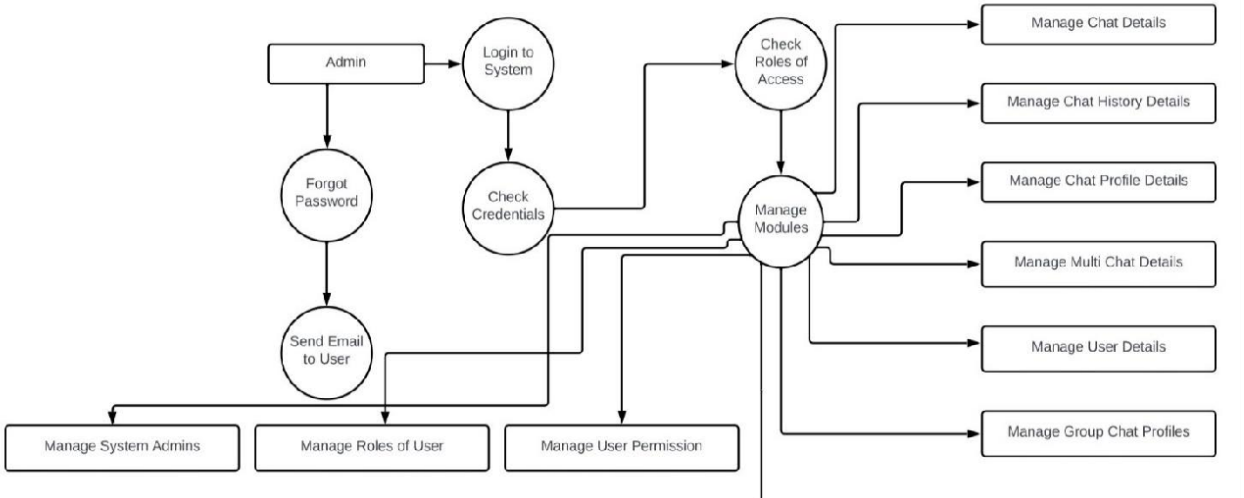
Zero Level DFD-Multi User Chat Applicaions

1 Level DFD



First Level DFD - Multi User Chat Application

2 Level DFD



Second Level DFD-Multi User Chat Applicaions

REFERENCES

Books:

1. Full-Stack
2. Modern Full-Stack DevelopmentPro MERN Stack
3. React Basic
4. The Road to Learn ReactReact Explained
5. React Projects

Websites:

1. https://www.w3schools.com/
2. https://getbootstrap.com/
3. https://reactjs.org/
4. https://www.google.com/

Faculty Guidelines:

Mr. Akash Kumar Choudhary (Technical Trainer , GLA University)

GitHub Repository link:

https://github.com/sourabhthakur87/Real-Time-Chat-App