MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Gov. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Department of Mathematics and Computing

2023-2024

MINOR PROJECT REPORT ON

CHAT WEBSITE

A Minor Project

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

ENGINEERING MATHEMATICS & COMPUTING

Submitted by

Yash Mathe (0901MC211068), Aakash Sharma(0901MC211008),
Abhishek Patel(0901MC211004),Nishant Dhakad(0901MC2211043)

Under the Guidance of

Prof. Prabhakar Sharma

DEPARTMENT OF ENGINEERING MATHEMATICS & COMPUTING

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE

GWALIOR, 474005 M.P., INDIA

Undertaking

Wehereby declare thatthe work presented in this project entitled "***Chat website***" submitted to the Department ofEngineering Mathematics and Computing, Madhav Institute of Technology and Science Gwalior, for the partial fulfilment of the requirements of the Bachelor of Technology degree in Engineering Mathematics and Computing. We further declare that this work has not beenthe basis for the award of any other degree, diploma or any other title elsewhere.

Date: 22/11/2023

Yash Mathe(0901MC211068)
Aakash Sharma(0901MC211008)
Abhishek Patel(0901MC211068)
Nishant Dhakad(0901MC211043)

Place: Gwalior

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE GWALIOR

Certificate

This to certify that dissertation entitled **"Chat Website"** which is being submitted by Yash Mathe (0901MC211068), Aakash Sharma(0901MC211008), Abhishek Patel(0901MC211068), Nishant Dhakad(0901MC211043)for the award of degree of **Bachelor of Technology** degree in Engineering Mathematics and Computing, MITS Gwalior is a record of benefited work carried out by them under my supervision. This dissertation has reached the standard fulfilling the requirements of the regulations relating to the degree.

23/11/2023

Prof. Prabhakar Sharma

Department of Engineering Mathematics and Computing

Madhav Institute of Technology and Science Gwalior

Acknowledgement

It is our great pleasure to express sincere gratitude to my supervisor, **Prof. Prabhakar Sharma**for his expert guidance and constant encouragement. We acknowledge that it is because of his interest that we enjoyed working on this project and express my earnest and heartfelt thanks to him for his time, support and efforts.

We are also thankful to all the faculties of the **Department of Engineering Mathematics and Computing** for their encouragement, who had invested their valuable time in providing their feedback with a lot of useful suggestions.

We are highly obliged to all my friends for their encouragement and for helping me at the points where I got stuck. I am deeply indebted to all of them for always helping and inspiriting me.

Yash Mathe (0901MC211068)
Aakash Sharma(0901MC211008)
Abhishek Patel(0901MC211068)
Nishant Dhakad(0901MC211043)

AKASH SHARMA
NISHANT DHAKAD
YASH MATHE
ABHISHEK PATEL

INDEX

Introduction

1.1 Purpose:

Utilizing technology to overcome geographical barriers, chatting serves as a means of bringing people and ideas together. A chat application facilitates seamless communication across the globe by enabling the real-time exchange of messages. Whether accessed through a web browser or mobile device, users can enjoy engaging and lively interactions through personalized messaging features, mimicking the spontaneity of in-person conversations. This not only fosters a sense of connection but also encourages users to continue their conversations on your platform rather than seeking alternative messaging solutions. To initiate chatting, the client establishes a connection with the server, where private conversations can take place.

1.2. Overview

1. **Real-Time Communication:** Implementing a robust real-time communication system to ensure instant message delivery.
2. **User Authentication:**Implementing secure user authentication mechanisms to safeguard user data and privacy.
3. **Intuitive User Interface:**Crafting an interface that is both intuitive and visually appealing, thereby elevating the overall user experience.
4. **Technological Proficiency:** Leveraging the MERN stack (MongoDB, Express.js, React.js, Node.js) to harness the strengths of each technology for a cohesive and efficient development process.

1.3. Motivation

The project is motivated by the need to tackle contemporary need for effective and user-friendly communication tools. As the world becomes more interconnected, the importance of platforms that foster instant communication has never been greater. Whether for casual conversations, collaboration among team members, or connecting with friends and family, a well-designed chat website serves as a central hub for diverse social interactions.

1.4 Technology Stack

- MongoDB: Document database for storing chat messages.

- Express.js: Web application framework for building the server.
- React.js: A front-end library designed for constructing the user interface.
- Node.js: JavaScript runtime for server-side development.

1.5Why MERN Stack

- **Unified Language (JavaScript):**
  - **Consistency:** The entire stack is written in JavaScript, promoting a consistent language across both frontend and backend development.
- **React.js for Dynamic UI:**
  - **Efficient Frontend:** React.js enables the creation of modular and interactive user interfaces, enhancing the overall user experience.
- **Node.js for Asynchronous Backend:**
  - **Scalability:** Node.js' asynchronous, non-blocking nature is well-suited for handling multiple concurrent connections, crucial for real-time applications like chat.
- **Express.js for Backend Structure:**
  - **Simplicity:** Express.js is a lightweight and fast framework, providing a straightforward structure for building the backend of the application.
- **MongoDB for Flexible Data Storage:**
  - **Adaptability:**The schema-less design of MongoDB enablesflexible data storage, accommodating the varied formats of chat messages.
- **Real-Time Capabilities with WebSockets:**
  - **Instant Updates:** Integration of WebSocket technology (e.g., Socket.io) enables real-time bidirectional communication, ensuring instant message updates.
- **Community Support and Documentation:**
  - **Vibrant Community:** The MERN stack has a large and active community, offering support, resources, and solutions to common challenges.
  - **Comprehensive Documentation:** Each component of the stack is well-documented, making it easier for developers to understand and implement features.

System Design

Client:

The user interface is represented by the chat client, whether it's a desktop, web, or smartphone application. This component is responsible for engaging with the operating system and handles tasks such as message interactions, data display for the user, and message storage. When a user sends a message, the chat client transmits it to the other key component: the chat server.

Server:

The chat server is a hosting platform for all the necessary software, frameworks, and databases required for the chat application to function. Its role involves receiving messages from the client side, identifying the correct sender, queuing the message, and forwarding it to the recipient's chat client. Typically, the chat server's resources encompass a REST API, a WebSocket server, and a database instance for message storage.

Socket.io:

Socket.io is a JavaScript library facilitating real-time, bidirectional communication between clients (such as web browsers) and servers. Widely employed for applications demanding instantaneous updates, it is commonly used in the development of chat applications, online gaming platforms, and collaborative tools.

Chat REST API:

The Chat REST API serves as an interface utilizing the REST protocol for building an online chat website. This protocol, constituting the architectural style, enables the app to send and receive requests and responses using HTTP methods like GET, POST, PUT, and DELETE. The API establishes specific endpoints (URLs) to access various chat functionalities, allowing HTTP requests like GET and POST to execute operations such as sending messages and retrieving message history.

3.Database

Our chat application, ChatApp, utilizes MongoDB as its database solution to efficiently manage user information, conversations, and messages. The database, named chatAppDB, is designed to ensure seamless communication and retrieval of data.

Collection: Users

The Users collection stores information about registered users:

- **Fields:**
  - _id (ObjectID): Unique identifier for each one.
  - username (String): Display name or username.
  - Email (String): It contain the email of users.
  - password (String): Securely hashed password for authentication.
  - createdAt (Date): Timestamp for user account creation.
  - avatarImage(string): Image for our profile picture.

Collection: Messages

The Messages collection is responsible for storing individual chat messages:

- **Fields:**
  - _id (ObjectID): Unique identifier for each message.
  - Users(ObjectID): Both user ID's
  - senderId (ObjectID): ID of the user sending the message.
  - content (String): Whatever message you write.
  - timestamp (Date): Timestamp for when a message is sent.

Relationships:

- **Conversations to Messages:**

  Each conversation can have multiple messages (one-to-many relationship).

- **Users to Conversations:**

  Each user can participate in multiple conversations (many-to-many relationship).

- **Messages to Users and Conversations:**

  Messages are linked to both conversations and users through their respective IDs.

User interface

Register

A register page, also commonly known as a registration page, is a web page that allows us to create a new account or register for the website. The primary purpose of a register page is to collect essential information from users, typically including personal details and credentials, to establish a unique account for them. Here are key components and functionalities on the register page:

- **User Information:**
  - **Username/Full Name:** Users often need to provide a unique username or their full name.
  - **Email Address:** A valid email address is usually required for communication and account verification purposes.
  - **Password:** Users are asked to create a secure password to protect their account.
  - **ConfirmPassword:** To check the password again.
- **Login Link:**
  - **Convenience:** Register page includes a link for users to log in immediately after registering, providing a seamless transition to the logged-in state.

Login

A login page is a web page that provides users with a secure entry point to access the website. Its primary function is to authenticate users by verifying their identity through valid credentials, such as a username and password. Here are key components and functionalities on the login page:

Credential Fields:

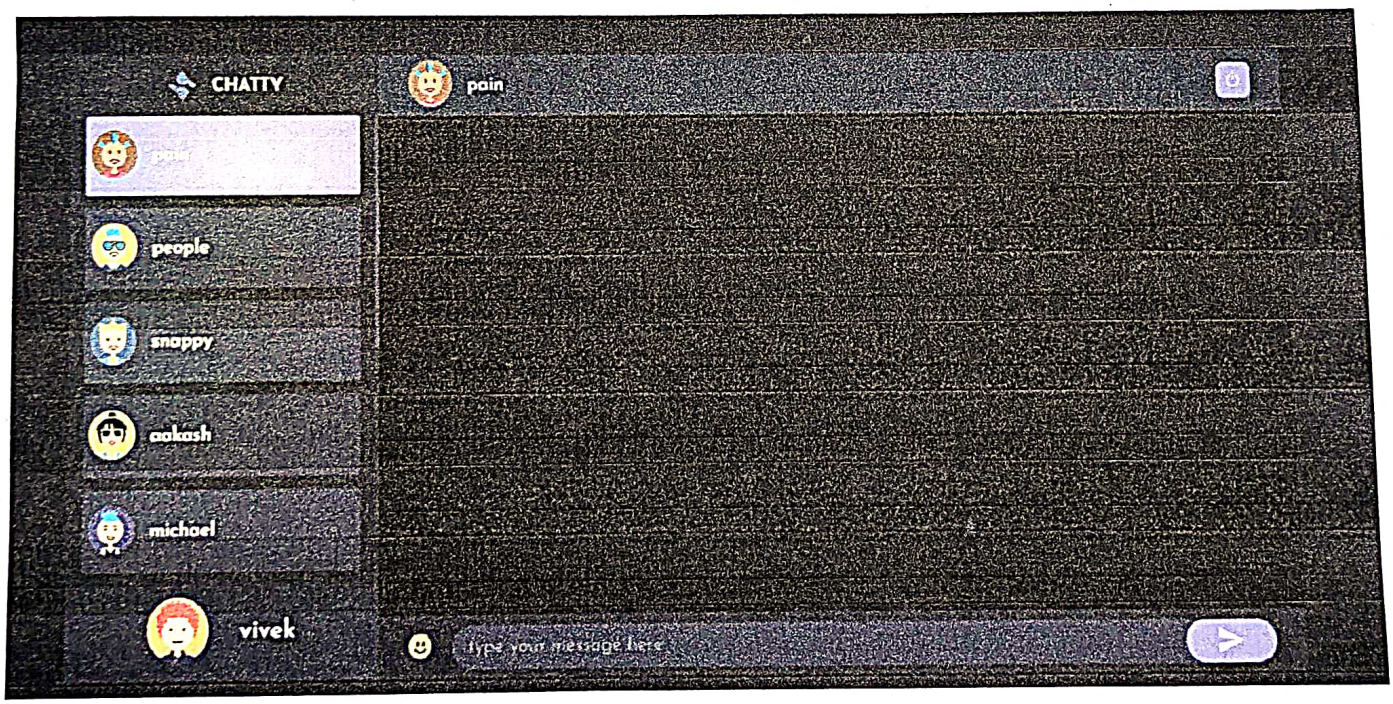
- **Username:** Users enter their unique username or the email address associated with their account.
- **Password:** A secure password is required for account authentication.

Chat:

A chat page is a user interface within a web application that facilitates real-time communication between users. It serves as a platform for sending and receiving messages, engaging in conversations, and includes additional features to enhance the user experience. Here are key components and functionalities on the chat page:

1. **Message Display Area:**
    - **Message Thread:** A visual representation of the ongoing conversation, displaying messages in chronological order.
    - **User Avatars:** Profile pictures or avatars associated with each user to identify message senders.
2. **Message Input Box:**
    - **Text Input Field:** A box where users can type and send messages.
    - **Emoji Support:** Optional functionality for users to include emojis in their messages.
    -
3. **Contacts:**

.**Users**: It Contains all the user that has created an account.



Challenges faced

- **Scalability:** As user numbers grow, ensuring that the chat application remains scalable is a significant challenge. Handling a large number of concurrent users and messages requires a robust backend infrastructure to prevent performance issues.
- **Real-time Communication:** Achieving low-latency and real-time communication is crucial for a chat application. Synchronizing messages across various devices and platforms in real-time without delays or disruptions can be challenging.
- **Cross-Platform Compatibility:** Ensuring a consistent user experience across different operating systems (Android, web) and devices can be challenging. Each platform may have unique requirements and design guidelines.
- **Security Concerns:** Security is a top priority for chat applications, especially when handling sensitive information. End-to-end encryption, secure authentication, and protection against various forms of cyber threats are essential components but can be challenging to implement effectively.

User Authentication and Authorization: Managing user authentication and authorization securely is crucial to prevent unauthorized access to sensitive conversations. It involves implementing secure login processes, protecting user credentials, and defining appropriate access controls.

- **Data Storage and Management:** Efficiently storing and managing vast amounts of chat data, including messages, multimedia files, and user metadata, can be challenging. This includes designing a database schema that supports fast retrieval and updates.
- **User Experience Design:** Designing an lucrative and users friendly interface that caters to a diverse user base can be challenging. Balancing simplicity with feature-rich functionality while maintaining an aesthetically pleasing design is an ongoing challenge.
- **Testing and Quality Assurance:** Thoroughly testing the application across different devices, operating systems, and network conditions is essential to identify and address bugs, performance issues, and ensure a seamless user experience.

Conclusion

In conclusion, developing a chat application offers numerous benefits, facilitating seamless communication and collaboration in both personal and professional settings. With the proliferation of smartphones and the increasing need for instant communication, a well-designed chat application can become an indispensable tool.

The key features of a successful chat application include user-friendly interfaces, real-time messaging, multimedia support, and robust security measures to ensure user privacy. Additionally, incorporating features such as group chats, voice and video calls, and integration with other platforms can enhance the overall user experience.

As technology continues to evolve, future developments in chat applications may contains the integration of artificial intelligence for improved chatbot interactions, advanced language processing capabilities, and personalized user experiences. Moreover, staying abreast of security protocols and addressing potential vulnerabilities will be crucial to maintaining user trust and safeguarding sensitive information.

Ultimately, a well-executed chat application has the potential to revolutionize the way people communicate, fostering connections across the globe and streamlining information exchange. Whether for social interactions, business collaborations, or customer support, a robust chat application can be a powerful tool in the digital age.

turnitin

Similarity Report ID: oid:28506:47242236

PAPER NAME

Word File Minor project Sample(2).docx

WORD COUNT

1673 Words

CHARACTER COUNT

10518 Characters

PAGE COUNT

9 Pages

FILE SIZE

143.2KB

SUBMISSION DATE

Nov 23, 2023 5:16 PM GMT+5:30

REPORT DATE

Nov 23, 2023 5:16 PM GMT+5:30

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 0% Internet database
- Crossref database
- 6% Submitted Works database
- 0% Publications database
- Crossref Posted Content database

Excluded from Similarity Report

- Bibliographic material
- Cited material

Summary