The Pharma Innovation Journal 2019; 8(3): 586-589

www.ThePharmaJournal.com

The Pharma Innovation

ISSN (E): 2277- 7695
ISSN (P): 2349-8242
NAAS Rating: 5.03
TPI 2019; 8(3): 586-589
© 2019 TPI
www.thepharmajournal.com
Received: 13-01-2019
Accepted: 15-02-2019

Dr. Avadhesh Kumar Gupta
Assistant Professor, Computer Science & Engineering, Lingaya's Vidyapeeth, Faridabad, Haryana, India

Correspondence Author:
Dr. Avadhesh Kumar Gupta
Assistant Professor, Computer Science & Engineering, Lingaya's Vidyapeeth, Faridabad, Haryana, India

Secured communication through a private chatting application

Dr. Avadhesh Kumar Gupta

DOI: https://doi.org/10.22271/tpi.2019.v8.i3j.25397

Abstract
Web based applications like private chatting applications have gained tremendous popularity in recent time. Due to which one can see variety of private chatting applications with different features provided. In this paper a survey of recent advances in the field as well as different industries affected by it. Key issues of such application are presented with issues in it. Summary of research results of different ways to create a private chatting application are summarized in this research paper. Advantages and drawbacks of the discussed system are discussed finally.
In our project we have created a one to one and a group chat application which provides user with enough freedom to customize interface to the liking. Making sure of user's privacy we have also tried to provide encryption to ensure his/her chat security. We have also after researching found several interesting ideas whom we have implemented in it like stories updates, light/dark mode for switching as well as custom multiple colors preset. Also we will incorporate audio calling as well as video calling to enhance its function and to make it as much as fulfilling to the user.

Keywords: Web based application, web development, backend development, frontend development, encryption

Introduction
How fast can one connect with another person has always been important. The advent of the internet has revolutionized the communication sector. Through this in early 90s it was possible to communicate between two computers. And the invention of smartphone has taken it to next level as people are now able to communicate with people no matter how far they are in an instant. In today's fast paced world, people prefer instant communication. The need of web applications like WhatsApp comes from this [1].
In the 1990s, the emergence of email and the World Wide Web (WWW) revolutionized the way people communicate and access information. Email provided a faster, cheaper, and more convenient way of sending messages than traditional mail or fax [2]. The WWW initially created as a platform for sharing research documents, quickly evolved into a global network of interconnected web pages, enabling users to access and share information on a vast range of topics and formats [3].
The early 2000s saw the rise of instant messaging (IM) services such as ICQ, AOL Instant Messenger, and MSN Messenger, which allowed users to send text messages in real time and engage in group chats. These services were later surpassed by mobile messaging applications such as WhatsApp, We Chat, and Facebook Messenger, which offered additional features such as voice and video calls, file sharing, and social networking integration [4].
The proliferation of mobile devices such as smartphones and tablets has further transformed communication, enabling users to access the internet, social media, and messaging services on the go. The rise of social media platforms such as Facebook, Twitter, and Instagram have also had a profound impact on communication, enabling users to connect with each other, share their thoughts and experiences, and participate in online communities [5].
Key components of such web-based applications are like User interface, Sever–side technology, Client-side technology, Messaging protocol, Authentication and security. The user interface is the part of the application that users interact with, and it includes features such as chat windows, message lists, and contact lists. The server-side technology handles the processing and storage of messages, user accounts, and other application data. The client-side technology runs on the user's device (e.g. web browser, mobile app), and is responsible for

displaying the user interface and handling user input [6]. This includes client-side scripting languages such as JavaScript, HTML, and CSS, which are used to create dynamic web pages and handle user interactions. The messaging protocol specifies the rules and format for sending and receiving messages between users [9]. Common messaging protocols used in web-based chatting applications include HTTP, WebSocket's, and XMPP. To ensure the security of user data and prevent unauthorized access to user accounts and messages, web-based chatting applications typically use authentication and security measures such as encryption, authentication tokens, and multi-factor authentication [7].

The remainder of this paper is organized as follows. In Chapter 2, we will present some of the research paper and literatures which provided us with the interesting ideas and knowledge. In Chapter 3, we provide a summary of our proposed web-based application that we are working on [12]. In Chapter 4, we describe the methods used in the study. In Chapter 5 we provided the results of our product. Finally, in Chapter 6 we conclude our research paper.

Literature Survey

"Security Issues in Web-Based Applications: A Survey" by N. Gupta and M. Sachdeva (2016). This paper presents a survey of security issues in web-based applications, including attacks, vulnerabilities, and defense mechanisms. "Design and Implementation of a Web-Based Collaborative Learning Application" by E. Hamouda and M. Jemni (2014) [10]. This paper describes the design and implementation of a web-based collaborative learning application that supports learners' interactions and activities. "Web-Based Applications: A Study of Critical Factors for Success" by W. Scott Spangler, Jr. and Nancy R. Mead (2008) [13]. This paper discusses the critical factors for success in developing web-based applications, such as usability, reliability, and security. "Web-Based Application Development: A Comparative Study of Agile and Traditional Approaches" by K. Ramesh and S. Kavitha (2010) [9]. This paper compares agile and traditional approaches to web-based application development and evaluates their effectiveness in terms of quality, productivity, and customer satisfaction. "Security Issues in Web-Based Applications" by J. Yan, H. Wang, and W. Shi. This paper examines the security issues in web-based applications, including the most common types of attacks and the best practices for mitigating them. "Web-Based Applications for Healthcare: A Systematic Review" by M. Pinto, F. Oliveira, and P. Sousa [15]. This paper provides a systematic review of web-based applications for healthcare, including their features, usability, and effectiveness in improving patient outcomes. "Web-Based Applications: The Benefits and Challenges" by A. Agarwal and S. Garg. This paper discusses the benefits and challenges of web-based applications, including their accessibility, scalability, and cost-effectiveness. "Web-Based Applications for E-Learning" by C. Liang and S. Liang. This paper explores the use of web-based applications for e-learning, including their effectiveness in enhancing learning outcomes, engagement, and collaboration.

Proposed system

This application will be one to one chat application and also a group chat application. We will try to incorporate audio calling and video calling feature. This will be a group chatting application with stories/Updates.

In this we will make sure of person's privacy and will try to give person as much freedom as required on the application [16]. In this we have light/Dark mode for switching as well as we will give custom multiple color presets. We will also provide various custom setting options. We will also give Authentication feature for login and register [8].

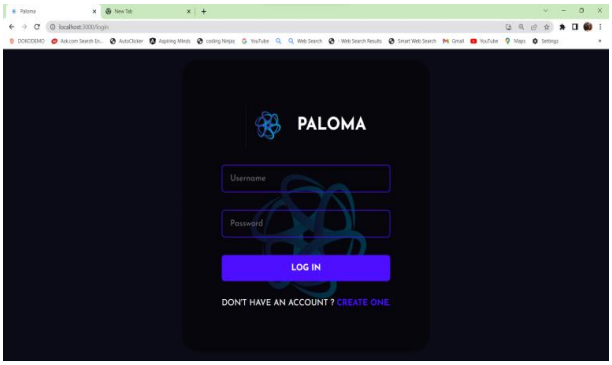

Fig 1: Login Page

Methodology

Developing a real-time chatting application involves several key steps that are crucial to ensure the system's reliability, security, and scalability [14]. Here is a high-level methodology for building a real-time chatting application:

1. Define the scope and requirements of the application: Determine the features and functionality of the application, including messaging capabilities, user management, security, and privacy requirements.
2. Choose the appropriate technology stack: Select the right programming language, database, and framework to build the application. Consider factors such as performance, scalability, and cost [11].
3. Design the architecture: Determine the system's architecture, including the front-end, back-end, and database. Choose a scalable and fault-tolerant architecture to ensure the system's reliability and uptime.
4. Develop the application: Build the application using the chosen technology stack and design. Implement features such as real-time messaging, user authentication, and data storage.
5. Test the application: Test the application for functionality, usability, security, and performance. Perform load testing to ensure the system can handle a high volume of users.
6. Deploy the application: Deploy the application to a production environment, such as a cloud-based server, and configure it for scalability and fault tolerance.
7. Monitor and maintain the application: Monitor the application for performance issues, security vulnerabilities, and scalability constraints. Regularly update the application with bug fixes and new features.

Here are the various languages required for doing so:

- **JavaScript:** JavaScript is a popular language used for building web applications, including real-time chatting applications. It is used for both the front-end and back-end development of the application.
- **Node.js:** Node.js is a server-side JavaScript runtime environment that can be used to build real-time chatting applications. It provides a fast and scalable platform for building server-side applications.
- **HTML/CSS:** HTML and CSS are used for designing and

styling the front-end user interface of the real-time chatting application.

- **WebSocket:** WebSocket is a protocol used for real-time communication between the client and server. It allows for bi-directional communication and is commonly used for real-time chatting applications.
- **PHP:** PHP is a server-side scripting language that can be used to build real-time chatting applications. It is commonly used in conjunction with WebSocket and other real-time technologies.
- **Redux:** Redux is a state management library for JavaScript applications, commonly used with React. It was developed by Dan Abramov and Andrew Clark and was inspired by Flux, another state management library. Redux helps manage the state of an application by providing a centralized store for all application data. This allows for a predictable and consistent approach to handling application state, and helps avoid the common pitfalls of shared mutable state.
- **MongoDB:** MongoDB is a popular cross-platform document-oriented database program that uses a NoSQL (non-relational) approach to store data. It is designed to be scalable, flexible, and high-performance, making it a popular choice for modern web applications. Instead of storing data in tables and rows like traditional relational databases, MongoDB stores data in flexible JSON-like documents, which can have fields and values of various types. This allows for greater flexibility and scalability, as well as faster query speeds.
- **Evaluation:** You can evaluate the performance of the chatting application with performance metric like security, reliability, speed, scalability, user experience and compability.

Results

In this section we will describe how much of the part of the proposed system we have achieved. In the final product we have achieved a user-friendly interface which is easy to use and pleasing to eye.

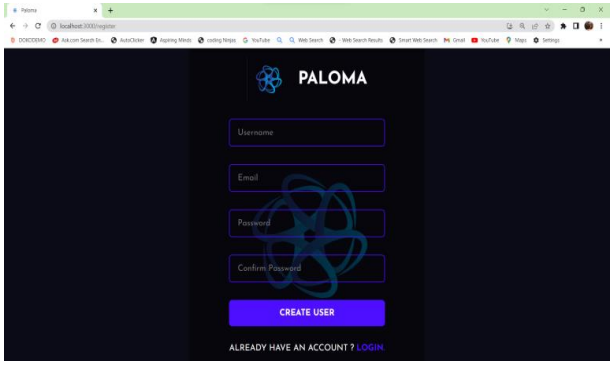

Fig 2: Registration Page for new user

Also we were also able to provide the basic requirements for a private chatting application like showing online status of a person as well as sending various type of media. We also provided group chat option.

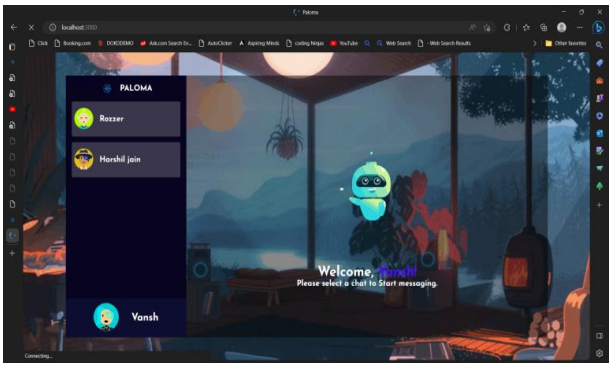

Fig 3: UI of Application

Conclusion

In conclusion, a real-time chatting application can be a valuable tool for individuals and businesses alike. The experimental setup for such an application involves several key components, including a powerful server, a robust database, web sockets, a reliable front-end application, authentication, and testing to ensure that the system can handle heavy traffic without delay or lag.

By leveraging these components and designing a wellarchitected system, real-time chatting applications can provide users with a seamless and responsive experience that enhances communication and collaboration. With the right tools and expertise, developing and deploying a realtime chatting application can be a rewarding project that helps to bring people together and facilitate more efficient communication.

References

1. "A Framework for Building Web-Based Educational Applications" by R. Baeza-Yates and C. Castillo. This paper presents a framework for building web-based educational applications, including the design principles, architecture, and implementation details.
2. "A Survey of Web-Based Applications and Tools for Data Visualization" by E. Papoutsaki, M. Petri, and N. Polyzotis. This paper surveys various web-based tools and applications for data visualization, including their features and limitations.
3. "Design and Implementation of a Web-Based Collaborative Learning Application" by E. Hamouda and M. Jemni (2014). This paper describes the design and implementation of a web-based collaborative learning application that supports learners' interactions and activities
4. "Security Issues in Web-Based Applications" by J. Yan, H. Wang, and W. Shi. This paper examines the security issues in web-based applications, including the most common types of attacks and the best practices for mitigating them.
5. "Security Issues in Web-Based Applications: A Survey" by N. Gupta and M. Sachdeva (2016). This paper presents a survey of security issues in web-based applications, including attacks, vulnerabilities, and defense mechanisms.
6. "Web-Based Application Development: A Comparative Study of Agile and Traditional Approaches" by K. Ramesh and S. Kavitha (2010). This paper compares agile and traditional approaches to web-based application development and evaluates their effectiveness in terms of quality, productivity, and customer satisfaction.

7. "Web-Based Applications for Business Process Management" by M. Dumas, M. La Rosa, and J. Mendling. This paper examines the use of web-based applications for business process management, including their role in supporting collaboration, automation, and integration.
8. "Web-Based Applications for Collaborative Learning" by J. Bransford, A. Brown, and R. Cocking. This paper discusses the use of web-based applications for collaborative learning, including their benefits and challenges.
9. "Web-Based Applications for Disaster Management" by S. Dutta and A. Pal. This paper presents an overview of web-based applications for disaster management, including their use cases, features, and limitations.
10. "Web-Based Applications for E-Learning" by C. Liang and S. Liang This paper explores the use of web-based applications for e-learning, including their effectiveness in enhancing learning outcomes, engagement, and collaboration.
11. "Web-Based Applications for E-Learning" by C. Liang and S. Liang. This paper explores the use of web-based applications for e-learning, including their effectiveness in enhancing learning outcomes, engagement, and collaboration.
12. "Web-Based Applications for Healthcare: A Systematic Review" by M. Pinto, F. Oliveira, and P. Sousa.This paper provides a systematic review of web-based applications for healthcare, including their features, usability, and effectiveness in improving patient outcomes
13. "Web-Based Applications for Personalized Medicine" by A. Madhavan, E. Denny, and L. Rosenfeld. This paper discusses the use of web-based applications for personalized medicine, including their role in enabling patient-centric healthcare, data analytics, and decision support.
14. "Web-Based Applications: A Study of Critical Factors for Success" by W. Scott Spangler, Jr. and Nancy R. Mead (2008). This paper discusses the critical factors for success in developing web-based applications, such as usability, reliability, and security.
15. "Web-Based Applications: The Benefits and Challenges" by A. Agarwal and S. Garg. This paper discusses the benefits and challenges of web-based applications, including their accessibility, scalability, and cost-effectiveness.
16. Kaushik P, Yadav R Reliability design protocol and block chain locating technique for mobile agent Journal of Advances in Science and Technology (JAST). 2017;14(1):136-141. https://doi.org/10.29070/JAST