

Tribhuvan University

Faculty of Humanities and Social Sciences

“Web Chat Application”

A PROJECT REPORT

Submitted to

Department of Computer Application

Lumbini City College

*In partial fulfillment of the requirements for the Bachelors in Computer Application*

Submitted by

Anup Rimal

(T.U.Reg.No: 6-2-1134-37-2020)

Bhupendra Thapa Chhetri

(T.U.Reg.No: 6-2-1134-38-2020)

Under the Supervision of

Er. Sujan Dhakal

Table of Contents

[Chapter: 1 Introduction 1](#_Toc172218399)

[1.1 Introduction 1](#_Toc172218400)

[1.2 Problem Statement 1](#_Toc172218401)

[1.3 Objective 1](#_Toc172218402)

[Chapter: 2 Methodology 2](#_Toc172218403)

[2.1 Requirement Identification 2](#_Toc172218404)

[2.2 Feasibility Study 2](#_Toc172218405)

[2.3 High-Level Design of System: 3](#_Toc172218406)

[Chapter: 3 Gantt Chart 7](#_Toc172218407)

[Chapter: 4 Expected Outcome 8](#_Toc172218408)

[References 9](#_Toc172218409)

# Chapter: 1 Introduction

## 1.1 Introduction

A web chat application is like a digital space where people can have conversations over the internet in real-time. Think of it as a virtual room where users can type messages and instantly send them to others who are also logged in. These applications are often accessed through a web browser, making them easy to use without needing to download any additional software.

To get started, users typically sign up for an account or log in with existing credentials. Once logged in, they can see a list of contacts or join different chat rooms based on their interests or affiliations. Within these chat spaces, they can engage in one-on-one conversations or participate in group discussions with multiple people.

Web chat applications often come with various features to enhance the chatting experience. This may include emojis and stickers to express emotions, file sharing capabilities to exchange documents or images, and even voice or video calling options for more dynamic communication. Additionally, some platforms offer customization settings, allowing users to personalize their profiles or chat environments to suit their preferences.

## 1.2 Problem Statement

Here's a simplified problem statement for a web chat application:

1. User Authentication: The application must provide secure user authentication, ensuring only registered users can access the chat features.
2. Real-Time Messaging: Implement real-time messaging to allow users to send and receive messages instantly without needing to refresh the page.
3. Scalability: The system should be designed to handle a large number of concurrent users and messages without performance degradation.
4. User Interface: Develop an intuitive and user-friendly interface that works seamlessly across various devices and screen sizes.
5. Data Security and Privacy: Ensure that all communications are encrypted and user data is stored securely, complying with relevant data protection regulations.

## 1.3 Objective

Here are the objectives of a web chat application:

1. To provide a secure user authentication system.
2. To implement real-time messaging capabilities.
3. To develop a scalable system for handling high traffic.
4. To provide user interface for communication.
5. To ensure data security and privacy compliance.

# Chapter: 2 Methodology

## 2.1 Requirement Identification

1. Study of Existing System:

The study of existing web chat applications such as WhatsApp and Microsoft Teams reveal several strengths and areas for improvement. These applications offer features like real-time messaging, multimedia support, and group chats, which are essential for modern communication. However, issues such as latency, limited scalability, and varying levels of security and privacy controls are prevalent. While some applications excel in user interface design and user experience, others may fall short in providing seamless cross-platform compatibility and robust encryption standards. By analyzing these aspects, we can identify the best practices to adopt and the gaps to address in our proposed web chat application to ensure a superior user experience and enhanced security.

1. Literature Review:

The literature review for the web chat application development involves examining various scholarly articles, industry reports, and technical documents to gather insights on current technologies and best practices. Key findings indicate that existing applications often face issues related to latency, scalability, and security. Studies highlight the importance of real-time data transmission protocols like Web Sockets, encryption methods for data security, and user interface design principles to enhance user experience. Additionally, the review underscores the necessity of cross-platform compatibility to cater to a diverse user base. These insights provide a solid foundation for developing a web chat application that addresses these critical areas effectively.

1. Requirement Analysis:

The requirement analysis for the web chat application involves gathering detailed information on user needs and preferences through various methods such as interviews, surveys, and focus group discussions. By understanding what features users find essential like real-time messaging, secure data transmission, and an intuitive interface we can define the core functionalities that our application must have. This process ensures that the final product will meet user expectations and address any existing shortcomings in current web chat applications.

## 2.2 Feasibility Study

1. Technical Feasibility:

An assessment of the technical resources, including hardware, software, and skilled personal required for the project, will be conducted. The study will also evaluate the technical challenges and potential solutions.

1. Operational Feasibility:

The operational feasibility study will evaluate the compatibility of the proposed system with existing workflows and processes. It will also assess the ease of use and the training requirements for end-users.

1. Economic Feasibility:

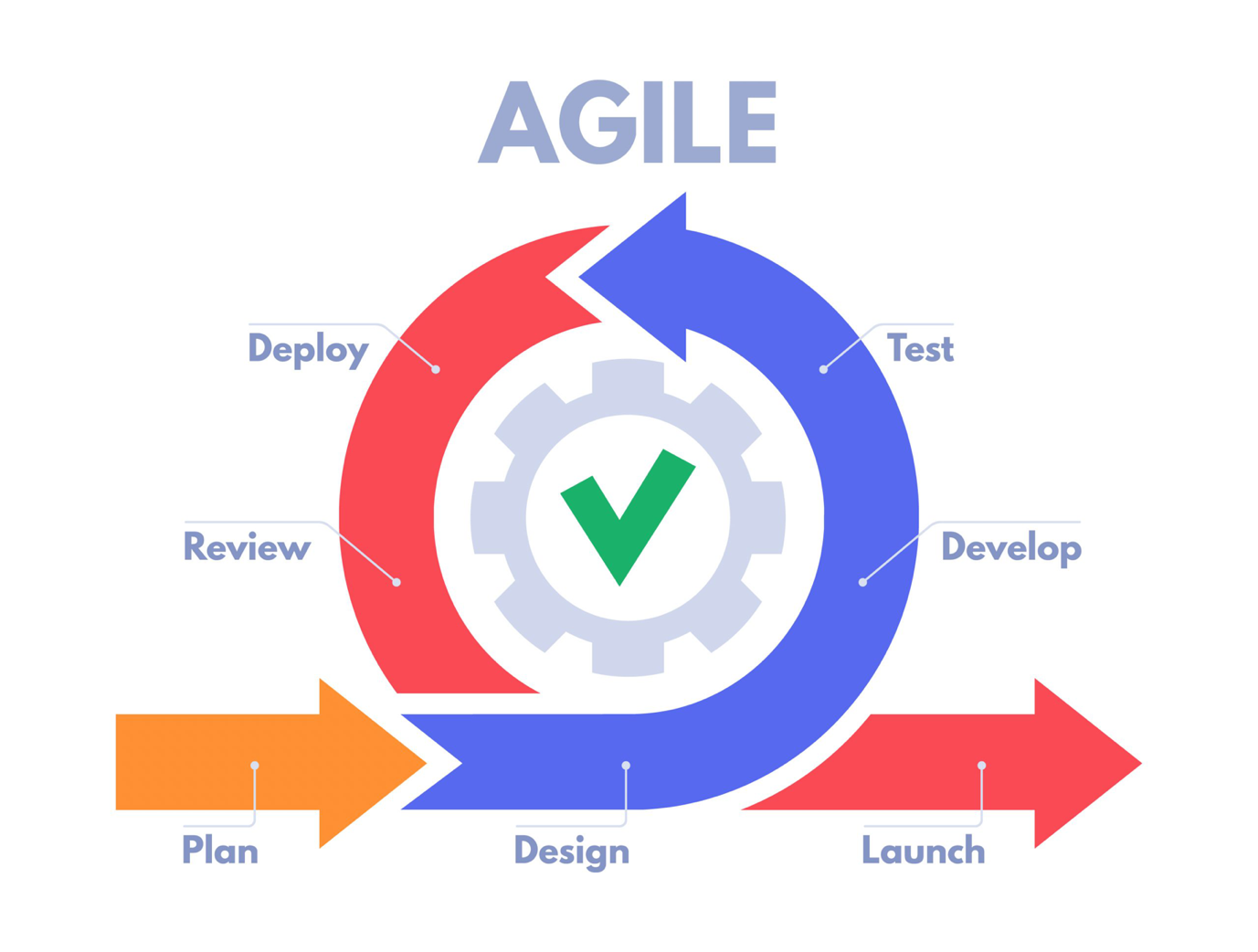
A cost-benefit analysis will be performed to evaluate the financial viability of the project. This will include the estimation of development costs, operational costs, and the expected return on investment (ROI).

## 2.3 High-Level Design of System:

1. Methodology of the Proposed System:

We are going to use the Agile methodology while building the web chat application. This project has specific documentation, time and requirements, well-understood technology so in order to build this system. Agile methodology can be used technology so in order to build this system. Here's how Agile principles can be applied to the development of a web chat application:

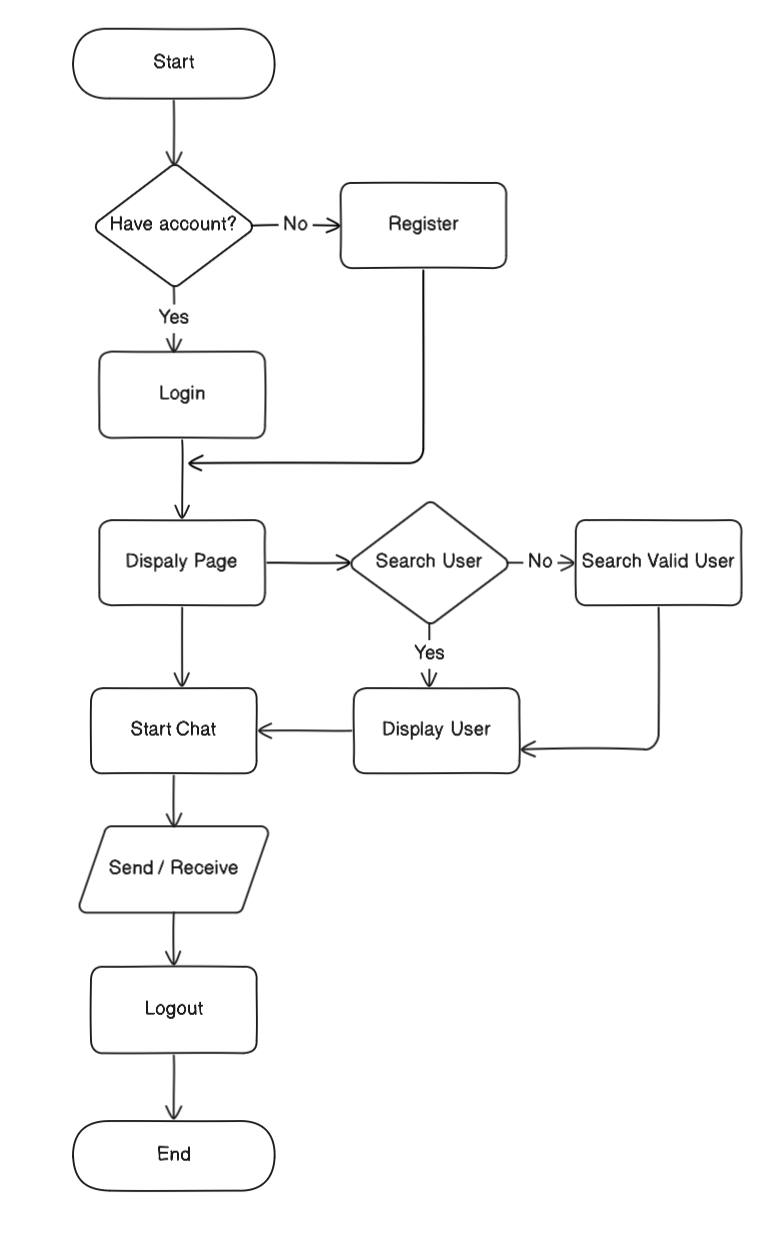
* Planning: We start by figuring out what needs to be done. We talk to the people who will use the chat app to understand what they want. Then, we make a list of all the things the app should do, like sending messages, creating profiles, and so on. We decide which things are most important and put them at the top of our list.
* Design: Once we know what the app should do, we think about how it will look and work. We draw pictures of each screen and think about where the buttons will go and how users will move around the app. We also decide how the app will store all the messages and user information.
* Development: This is where we actually build the app. We take each item from our list and work on it one at a time. For example, we might start by making a screen where users can log in. Once that's done, we move on to the next thing, like letting users send messages.
* Testing: After we've built something, we check to make sure it works correctly. We try to do things that might break the app, like typing in the wrong password or sending a really long message. If we find any problems, we fix them before moving on to the next thing.
* Deployment: Once everything works well and we're happy with it, we put the app on the internet so people can use it. This is called deployment. We make sure everything is set up correctly on the servers and that the app is easy for people to access.
* Review: After the app is live, we ask people to use it and tell us what they think. We pay attention to what they like and don't like, and if they find any problems. Then, we use that feedback to make the app even better. We might add new features or fix things that aren't working well.



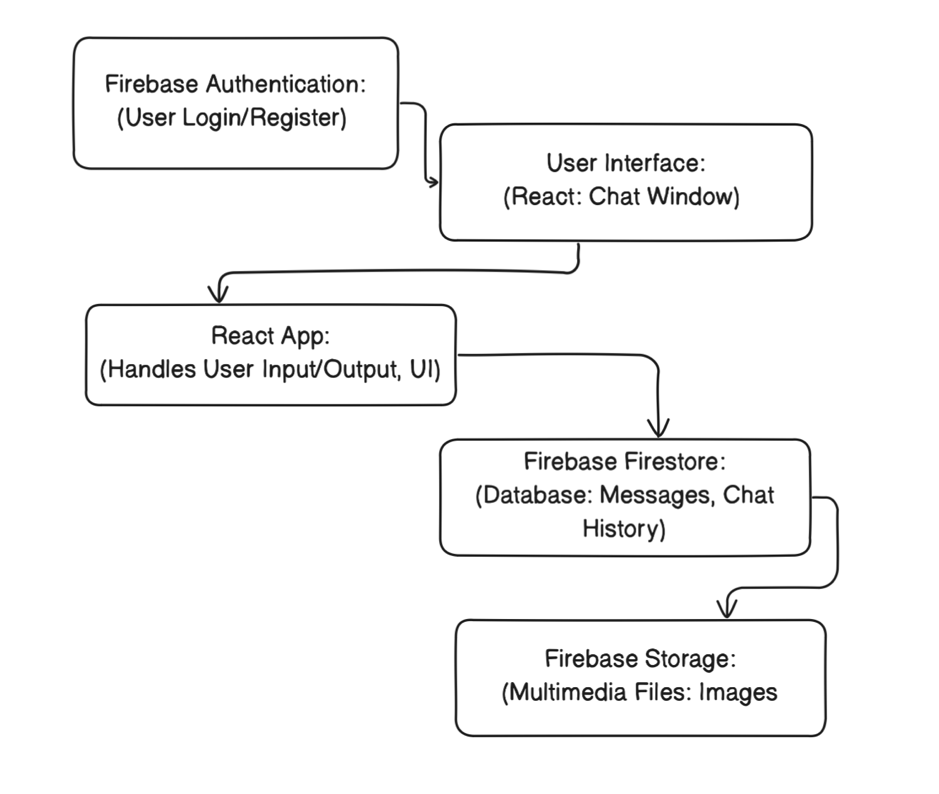
**Figure 1: Agile Methodology**

1. Flow Chart:

The flowchart illustrates the steps of a user interacting with a chat system. Here is a brief description: The process begins with the user determining if they have an existing account. If the user does not have an account, they proceed to the registration step. If they have an account, they log in. Upon successful login, the main page is displayed. The user can then search for another user to chat with. If a valid user is found, the system displays the user, and the chat session can be initiated. During the chat session, messages are sent and received. After finishing the chat, the user logs out, and the process ends. If no valid user is found during the search, the system prompts the user to search again. This will help in visualizing the data flow.



**Figure 2: Flow Chart**

1. Working Mechanism of Proposed System:
2. Description of Algorithms:

**Figure 3: Working Mechanism of System**

* User Authentication Algorithm:

Step 1- User Input: User enters email and password.

Step 2- Firebase Auth Request: The React app sends the credentials to Firebase Authentication.

Step 3- Validation: Firebase validates the credentials. If valid, Firebase returns a user token. If invalid, an error message is returned.

Step 4- User Session: The React app stores the user token in local storage or context for session management. Redirect the user to the chat interface upon successful authentication.

* Real-Time Messaging Algorithm:

Step 1- Message Input: User types a message and clicks send.

Step 2- Send Message: The React app sends the message to Firestore with user ID and message content. Firestore stores the message and triggers real-time updates.

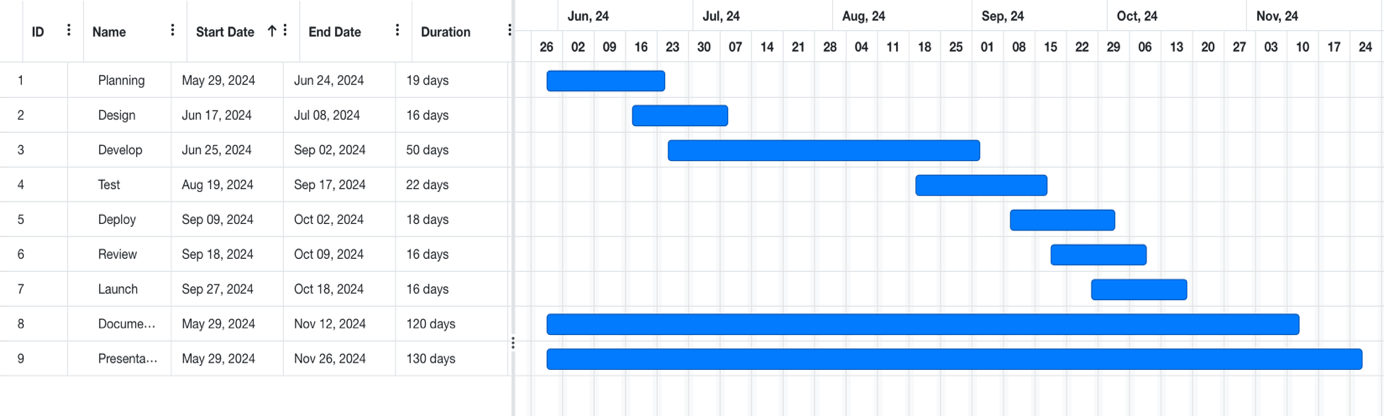
Step 3- Receive Message: Firestore updates all connected clients with the new message in real-time. The React app listens for these updates and updates the chat interface.

* Multimedia Support Algorithm:

Step 1- File Upload: User selects a file to send (image, video, etc.). The React app uploads the file to Firebase Storage. Firebase Storage returns the file URL. The React app stores the file URL in Firestore with the message metadata.

Step 2- File Retrieval: When displaying messages, the React app fetches the file URL from Firestore. The app uses the URL to display the multimedia content in the chat.

# Chapter: 3 Gantt Chart

A Gantt chart will be created to outline the project timeline, including key milestones and deliverables. This will provide a clear roadmap for the project and help in tracking progress.

**Figure 4: Gantt Chart**

# Chapter: 4 Expected Outcome

The expected outcome of the web chat application developed using React and Firebase includes a seamless real-time communication platform with robust user authentication and security features. Users will enjoy instantaneous message delivery in both one-on-one chats, supported by Firebase Authentication for secure access. Persistent chat history will be maintained in Firebase Firestore, allowing users to retrieve and review past conversations at any time. The application will support multimedia messaging, enabling the sharing of images, videos, and files, all securely stored in Firebase Storage. The user interface, built with React, will be intuitive and responsive, ensuring an excellent user experience across various devices and platforms. The application will be highly scalable and performant, thanks to Firebase's real-time database and serverless infrastructure. Additionally, robust security measures will protect user data and privacy, ensuring compliance with data protection regulations. Overall, this project aims to deliver a reliable, secure, and feature-rich web chat application that meets modern communication needs.

# References

|  |  |
| --- | --- |
| [1] | S. Dosani, "Real-Time Chat Application," *International Journal of Scientific Research in Science Engineering and Technology,* April 2023. |
| [2] | A. Sagar, "Real Time Chat Application using React and Firebase," *International Journal for Research in Applied Science and Engineering Technology,* April 2024. |
| [3] | K. Roy and M. Verma , Preview image for Development and Integration of AI-Powered Real-Time Chat Application Using OpenAI and Chat Engine APIs Source Development and Integration of AI-Powered Real-Time Chat Application Using OpenAI and Chat Engine APIs, June 2024. |
| [4] | P. Singh, "Encrypted Chat Application," *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT,* June 2024. |
| [5] | R. Gupta, "MySQL based Chit Chat Application," *International Journal for Research in Applied Science and Engineering Technology,* June 2024. |