SUNSTONE

CHAT APPLICATION

Task - 1

SUNSTONE

CHAT APPLICATION WEBSITE

- A chat application, often referred to as a messaging app, is a software or mobile application that enables users to exchange text, multimedia, and sometimes voice or video messages in real-time over the internet. These apps facilitate communication and conversation between individuals or groups, making it easy to connect and interact with others, regardless of geographical distance. Examples of popular chat applications include WhatsApp, Facebook Messenger, and Slack.

Task - 1

Creation of SRS & Github

- Create SRS :Context of mobile technology
- Creation & Set-up of Github account
- Creation & Hands-on to various commands of Git Bash

Evaluation Metric:

- 100% Completion of the above tasks

Learning Outcome

- Get to know about different lifecycle models.
- Understanding importance and how to create an SRS
- Knowing various commands of Github
- Understanding agile and scrum management techniques for efficient product development

Introduction

Our project is related to a new way of chatting with people. Chatting and communicating with people through internet is becoming common to people and is connecting people all over the world. Mainly, chatting apps in today's world mainly focus on connecting people, providing users with more features like GIFs, stickers etc. But this app, is different from them.

chat.

PURPOSE:

A Chat application's purpose is to facilitate real-time communication between users. It allows people to exchange text messages, multimedia, or even make voice and video calls, fostering instant and convenient conversations over the internet. Chat apps are used for various purposes, including personal communication, business collaboration, customer support, and more.

SOFTWARE REQUIREMENTS:

The software requirements for a chat application can vary depending on the specific features and platforms you want to target, but here are some common requirements:

Operating System:
Specify the target operating systems (e.g., Android, iOS, Windows, macOS, Linux).

Development Platform:
Choose the development framework or language (e.g., Java, Swift, React Native, Flutter) and the necessary development tools (e.g., Android Studio, Xcode, Visual Studio).

Backend Server:
You'll need a server to handle communication between users. Common choices include Node.js, Ruby on Rails, Python (Django or Flask), or .NET.

Database:
Select a database system (e.g., MySQL, PostgreSQL, MongoDB) to store user data and chat messages.

HARDWARE REQUIREMENTS:

The hardware requirements for a chat application will depend on various factors, including the scale of the application, expected user load, and the choice of infrastructure. Here are some general hardware considerations:

Server Hardware:
For small-scale applications, a single server may be sufficient. For larger applications, consider a cluster of servers.Multi-core processors with sufficient RAM for handling concurrent connections and processing messages quickly. Solid-state drives (SSDs) for faster data access.

Load Balancer:
If your application expects a high volume of users, consider load balancers to distribute incoming traffic across multiple servers for load balancing and redundancy.

Database Server:
A powerful server or cluster of servers for the database system to handle data storage and retrieval efficiently.

Networking:
High-speed internet connection with sufficient bandwidth to handle incoming and outgoing data traffic.

FUNCTIONAL REQUIREMENTS:

Functional requirements for a chat application outline the features and capabilities that the application must have to meet its intended purpose. Here are some common functional requirements for a chat application:

User Registration and Authentication:
Users should be able to register with a username and password or other authentication methods.
User authentication to ensure secure access to the chat application.

User Profiles:
Users can create and update their profiles, including profile pictures and status messages.

Contact List:
Users can manage a list of contacts and add or remove friends.

One-on-One Messaging:
Users can send text messages, emojis, images, and other multimedia to individual contacts.

Group Messaging:

Users can create and participate in group chats, sending messages to multiple users simultaneously.

NON-FUNCTIONAL REQUIREMENTS:

Nonfunctional requirements for chat applications are essential considerations that define how the application performs rather than its specific features. They are often related to the application's performance, security, scalability, and usability. Here are some common nonfunctional requirements for chat applications:

Performance:
Response Time: Messages should be sent and received with low latency for real-time communication.

Throughput:
The application should handle a high volume of concurrent users and messages.

Scalability:
The ability to scale to accommodate a growing user base.

Availability and Reliability:
The application should be available and responsive 24/7 with minimal downtime.
Redundancy and failover mechanisms should be in place to ensure reliability.

CHAT APPLICATION WEBSITE

Click Here to see the Project

Thank you!