****

SYNOPSIS

**ON**

**Simple Web-Based Chat Application**

Submitted By: Submitted To:

Anuj Chaudhary - D – 2115000172 Amrendra Singh

Dhairya Pandey - D – 2115000351 Assistant Professor

Vishal Singh - D - 2115001133 Coding Blocks

**Simple Web-Based Chat Application**

**Objective:**

* Develop a minimalist and user-friendly web interface for chatting.
* Implement real-time messaging using HTML, CSS, JavaScript
* Allow users to join different chat rooms for conversations.
* Ensure a clean and intuitive user experience for seamless communication.

**Scope:**

Inclusions:

* Real-Time Messaging
* Basic User Interface
* Hosting and Deployment
* Message History
* User Registration
* User Profiles

Exclusions:

* Multiple Chat Rooms (Single chat room only)
* Extensive User Management
* Additional Features

**Methodology:**

* Front-End: HTML, CSS, JavaScript
* Back-End: Node.js (or another server-side technology for handling real-time communication)
* Hosting: Deploy the application on a web server for accessibility

**Proposed System:**

The proposed system is a simple web-based chat application that provides a platform for real-time text-based communication among users. The core idea of this application is to create a minimalist and straightforward chat environment where users can join a single chat room and exchange messages in real time.

**Features:**

* Real-Time Messaging: Users can send and receive text messages in real time within the chat room.
* Single Chat Room: The application provides a single chat room for users to engage in conversations.
* User-Friendly Interface: The user interface is designed to be simple and intuitive, allowing for easy navigation and messaging.
* Hosting and Deployment: The chat application will be hosted on a web server, making it accessible from web browsers.
* Message History: Users can view and scroll through past messages within the chat room, providing context for ongoing conversations.
* User Registration: Users can create accounts by registering with a unique username.

**Implementation Plan:**

Week 1:

* Project Setup and Planning Define project objectives and requirements.
* Set up the development environment, including a web server.
* Create a project plan with a detailed schedule.

Week 2-3:

* Development Design a simple user interface for the chat application.
* Implement the front-end using HTML, CSS, and JavaScript.
* Develop a single chat room with real-time messaging functionality.

Week 4:

* Testing, Deployment, and Documentation Test the application for functionality and address any critical issues.
* Deploy the chat application on a web server for accessibility.
* Create minimal documentation with instructions for setup and basic usage.

**Team Members:**

Dhairya Pandey – Group Leader, Frontend

Anuj Chaudhary – Frontend

Vishal Singh - Backend

**Resources Required:**

Software:

* Text Editor or Integrated Development Environment (IDE): You'll need a code editor or IDE for writing HTML, CSS, JavaScript, and server-side code. Popular choices include Visual Studio Code, Sublime Text.
* Node.js: If you're using Node.js for the back end, you'll need to install Node.js and npm (Node Package Manager).
* Version Control: Git and a Git repository hosting service like GitHub or GitLab for managing the project's source code and version control.
* Documentation Tools: For creating project documentation, tools like Markdown editors or Google Docs can be helpful.

Hardware:

* Computer: A standard computer or laptop with internet connectivity is required for development.
* Web Hosting: To deploy your application, you may need a web hosting service or server. Depending on your choice, this can be a cloud-based server, shared hosting, or a dedicated server.

**References:**

MDN Web Docs, W3Schools, Node.js Documentation, Youtube, Github

**Expected Outcomes:**

By the end of the project, we expect to have a fully working chat application that people can use to talk to each other in real time on the internet. Users will be able to create accounts and log in, and the application will store the chat history so you can see previous messages. This application will be hosted on a web server, making it accessible to anyone with an internet connection.

**Conclusion:**

In conclusion, The core features of the application include real-time messaging, a single chat room, user registration and authentication, message history, hosting and deployment, and accompanying documentation. The project's primary goal is to provide a functional and minimalistic chat platform that offers a simple and enjoyable user experience. Success will be measured by the delivery of a fully operational chat application that adheres to the project's defined scope and requirements.