INT222:ADVANCED WEB DEVELOPMENT

SRS REPORT

on

REAL-TIME CHAT APPLICATION

School of Computer Science &Engineering

LOVELY PROFESSIONALUNIVERSITY

April 2024

SUBMITTED TO – DR. BRIJESH PANDEY

SUBMITTED BY:-

NAME	REG.NO	ROLL NO.
APOORVA SHUKLA	12210383	06

Software Requirements Specification (SRS) for Real-Time Chat Application

1. Introduction

1.1 Purpose

The purpose of this document is to outline the functional and non-functional requirements for developing a real-time chat application using Node.js, Express, and Socket.io. This chat application will allow users to register with a unique username, select chat rooms, engage in real-time messaging, and leave chat rooms as needed.

1.2 Scope

The chat application will enable users to:

- Register with a username and select a chat room.
- Join and participate in real-time messaging within selected chat rooms.
- Leave chat rooms when desired.

2. Overall Description

2.1 Product Perspective

The chat application will be a standalone web-based system where users can interact with each other in real-time. It will utilize Node.js and Express for server-side development and Socket.io for real-time communication between clients and the server.

2.2 Product Features

- User registration with a unique username.
- Chat room selection during registration.
- Real-time messaging within selected chat rooms.
- Ability to leave chat rooms.
- Notification system for new messages.
- User status indicators (online/offline).
- Profile management (update user information).

2.3 User Classes and Characteristics

- **Guest User:** Can access the login/register page.

- **Registered User:** Can log in, chat with others, create/join groups, and manage their profile.

2.4 Operating Environment

The application will be web-based and accessible via modern web browsers.

3. Functional Requirements

3.1 User Authentication

- **FR-1:** Users can register with a unique username and select a chat room during registration.
    - **FR-1.1:** Users provide a unique username during registration.
    - **FR-1.2:** Users select a chat room they want to join during registration.
    - **FR-1.3:** Usernames must be unique across the system.

3.2 Room Management

- **FR-2:** Users can join a specific chat room upon registration or after logging in.
    - **FR-2.1:** The system supports multiple chat rooms, each with a unique identifier and name.
    - **FR-2.2:** Users can see a list of available chat rooms during registration.
    - **FR-2.3:** Users can select a chat room to join during registration or change rooms after logging in.
- **FR-3:** Users can leave a chat room.
    - **FR-3.1:** Users have the option to leave their current chat room at any time.
    - **FR-3.2:** Upon leaving a room, the user should no longer receive messages from that room.

3.3 Real-time Chat

- **FR-4:** Once registered and logged in, users can participate in real-time messaging within their selected chat room.
    - **FR-4.1:** Users can send and receive messages within the chat room using Socket.io.
    - **FR-4.2:** Messages are visible to all users within the same chat room.

3.4 User Management

- **FR-5:** Users can edit their profile information (e.g., username, profile picture).
- **FR-6:** Users can change their password

4. Non-functional Requirements

4.1 Performance

- **NFR-1:** The application should handle a large number of concurrent users efficiently.
- **NFR-2:** Chat messages should be delivered instantly with minimal latency.

4.2 Security

- **NFR-3:** User authentication and password storage should follow best security practices.
- **NFR-4:** All communication between clients and the server should be encrypted (e.g., using HTTPS).

4.3 Usability

- **NFR-5:** The user interface should be intuitive and responsive.
- **NFR-6:** Support for different screen sizes (responsive design).

4.4 Technology Stack

- **NFR-7:** The application will be built using Node.js, Express, and Socket.io.

- **NFR-8:** Use a relational or NoSQL database for data storage (e.g., MongoDB, PostgreSQL).
-

5. Other Requirements

5.1 Legal and Compliance

- **OR-1:** The application must comply with relevant data protection laws (e.g., GDPR).
-

6. Appendix

6.1 Glossary

- **Socket.io:** A JavaScript library for real-time web applications.
- **Node.js:** A JavaScript runtime environment used for server-side applications.
- **Express:** A web application framework for Node.js

DFD (Data Flow Diagram) :