Android Real Time Chat application using Socket.io (Node JS)

Module 1: Introduction to Node.js and Basic Routing

- What is Node.js and its key features?
- Understanding the event-driven, non-blocking I/O model.
- Advantages of using Node.js for server-side development.
- Setting Up a Node.js Project: Installing Node.js and NPM.
- Creating a new Node.js project using npm init.
- Creating a Basic HTTP Server using the built-in http module.
- Handling HTTP requests and responses in Node.js.
- Introduction to Routing.

Module 2: Introduction to Socket.IO and Real-time Communication

- Overview of Socket.IO and its role in real-time communication.
- Understanding WebSocket and its advantages.
- Integrating Socket.IO with Node.js.

Module 3: Implementing the Server-side for the Chat App

- Setting up a Node.js server for the real-time chat application.
- Handling socket connections and disconnections.
- Implementing message broadcasting to all connected clients.

Module 4: Authentication with Android Application

- Prepare User Interface for Chat Application using Recycler View
- Implementing user authentication.
- Handling user login and registration requests.
- Securing the chat endpoints with authentication middleware.

Module 5: Real-time Message Handling

- Sending and receiving real-time messages using Socket.IO.
- Displaying received messages in the chat screen.
- Adding timestamps to messages.

Module 6: User Presence and Typing Indicators

- Displaying online/offline status of users.
- Implementing typing indicators.
- Notifying other users when a user is typing.

Module 7: Message History and Data Persistence

- Storing and managing message history.
- Implementing local data persistence for messages.
- Loading message history when entering a chat room.