Project Title	Audio Conference Platform
Technologies	MERN
Domain	Social Media
Project Level	Hard

Table

Table of Contents

1. Problem Statement:

Design a MERN stack-based Audio-Conferencing Platform

1.1. Overview of MERN Stack and Audio-Conferencing Platform

Full-stack web applications can be launched more quickly and easily with the JavaScript stack MERN. The MERN Stack is comprised of the four technologies MongoDB, Express, React, and Node.js. The goal is to simplify and streamline the development process.

Each of these four potent technologies provides programmers with a complete working environment and considerably aids in the creation of web applications.

We are all familiar with WhatsApp, a social networking software with several features that allows us to chat, call, and share with our friends. However, your task is to develop an audio-based social media application in which a user can communicate with his or her peers only through audio; he or she should not be able to share any text, image, video, or other media using the platform; audio will be the only available method of communication.

Note: One of the most well-known audio sharing platform apps is **Club House**, where you can find inspiration.

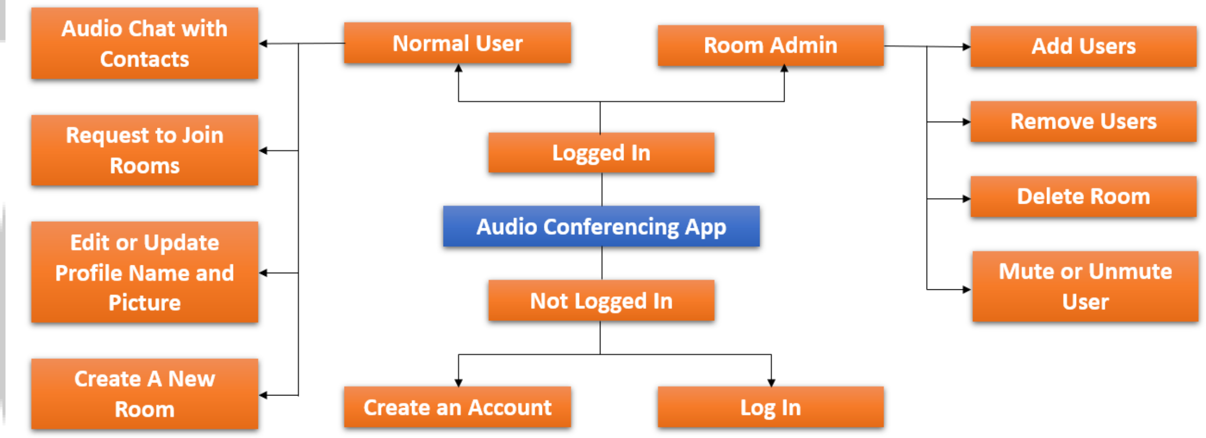
1.1.1 The Audio-Conferencing App should have the following properties.

These are some of the expected features of the application-

a. Strong authentication should be used so that users can register for an account or log in if they already are.

b. The user can use audio as a channel for communication with any linked contacts via the site.
c. The users can set up an audio chat room to directly communicate with one another through the group.
d. Users won't be permitted to communicate on the platform using any other media.
e. A user's name and profile photo may be displayed to other users of the platform.
f. Any user can ask to join another user's room, but he won't be included until the admin grants his request.
g. Any user may be added to or removed from the room by the admin.
h. Emoji and stickers can be used instead of voice or audio by users to speak privately or hang out in rooms.

Low Level user flow



1.2. Project Objective

• **Audio Conferencing App:** A unique experience is spending time with a loved one while listening to their voice and speaking to them verbally. No one has to reveal their face or waste time inputting text; instead, they can speak directly to one another and exchange opinions and knowledge.

• **User-friendliness:** The application should be user-friendly so that all types of users are at ease with it and with one another.

1.3. Scope of The Project

1. Nobody enjoys sending texts or just making a video call without revealing their face. Some of us find all these alternatives to be unsettling, but by using this platform, a user can hang out with a loved one without revealing their identity or wasting their time via messaging.

2. Simply gather in the family room to socialize with friends and family.

3. In the audio rooms, users can join and share their talents.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

1. **User Registration:** There must be a strong authentication system to prevent fraudulent actions. The user who has account can have access to the platform and no one should be able to create multiple account using same details. A user having an account can easily logged in to his account to get the access.

2. **Creating New Room:** A logged in user should be able to create his new audio room and can invite his peers to connect there.

1.4.2. Non-Functional Requirements

1. **Privacy:** A strong authentication system is required to secure all the existing user and to give assurance to new user.

2. **Robustness:** In case of user's system failure, a backup of his data should be there so that his rooms and personal details can be recovered easily.

3. **Performance:** The application should be light weight and optimized so that the user can easily communicate even on the low-speed internet connection.

1.5. Use Case Table

Authentication System	Register, Login, Logout	User
Search Form	Search Users	User
User Dashboard	Display all users and joined rooms	User

Table 1. Use Case

2. Project Evaluation Metrics:

2.1. Code:

- You are supposed to write code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

2.2. Database:
MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:
You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:
Implementation of reverse proxy, load balancing, and security group is mandatory for deployed applications.

2.5. Solutions Design:
You have to submit complete solution design strategies in High-level Document (HLD), Low-level Document (LLD), and Wireframe documents**.**

2.6. System Architecture:
You have to submit a system architecture design in your wireframe document and architecture document.

2.7. Optimization of solutions:
Try to optimize your solution on code level, architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:
You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:
You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.