

# Master Plan:

Laiba -> Agora integration.

Sohaib -> chat implementation.

Rimmel -> File server App.

Haris -> overall Frontend & SQA.

Hamza -> report & Technical help.

## Agora Itegration:

### Points to be known:

Agora.io is a real-time communication platform that provides voice and video APIs for developers to build web and mobile applications with high-quality audio and video functionality. It allows for live audio and video broadcasting, real-time communication, and interactive video conferencing. Agora also offers SDKs for various programming languages and platforms, including iOS, Android, Web, and Unity.

### Agora Integration:

Integration code mey base/views.py mey bhi aur base/static/stream.js mey bhi huwi hai.

Base/views/Stream.js dekhlo bs.

Iskey base/view.py mey 5 views hain

Jsimey phelay do basic page render krahay hain. Lobby wala view initial invoke hota hai , aur room wala view specific room mey direct krta hai.

Point to be known.

@csrf\_exempt

@csrf\_exempt is a decorator in the Django web framework that is used to exempt a specific view function or class-based view from the CSRF (Cross-Site Request Forgery) protection that is enabled by default in Django. This means that any requests to the decorated view will not be checked for a valid CSRF token, and will be processed as if the CSRF protection is not enabled. Django requires that a CSRF token be included in any form submissions or non-GET requests, to ensure that the request is coming from a trusted source.

baqi views ka function unkey name sey hi pata chaljayega. GET mem,create member,deletememeber.

## Sohaib Part:

### WEB Sockets

Sohaib tumhe yeh batana hai ky chat app implement kesay kari hai. Chat application ky liye hamlogon ney django channels ka use kra hai, Django Channels is an extension to the Django web framework that allows for the development of asynchronous, real-time functionality using WebSockets and other asynchronous protocols. It provides a way to handle and route WebSockets and other asynchronous protocols, such as HTTP/2, alongside standard synchronous HTTP requests in a Django application.

Django Channels is an extension to the Django web framework that allows for the development of asynchronous, real-time functionality using WebSockets and other asynchronous protocols. It provides a way to handle and route WebSockets and other asynchronous protocols, such as HTTP/2, alongside standard synchronous HTTP requests in a Django application.