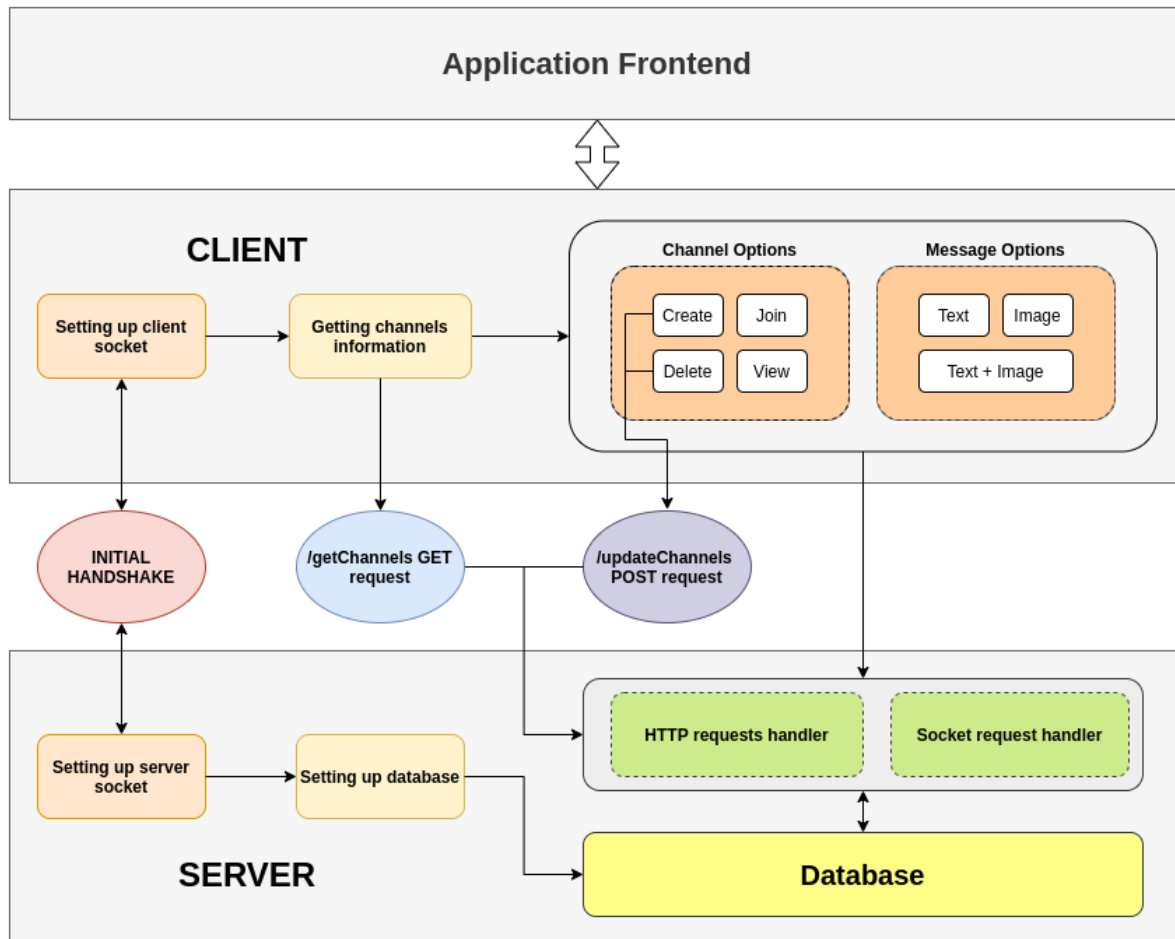


Problem Statement

A multi-client chat web application with image sending features.

Design description

Structural design:



Design Explanation:

1. The client and server is running on port 3000 and 8080 respectively.
2. The client and server sets up a socket and initial connection is done.
3. The client asks for channel information from the server using a HTTP GET request.
4. The client can create, join, view or delete a channel.
5. **CREATE** - The client creates a channel and updates the server about it using a HTTP POST request.
6. **JOIN** - The client joins a channel and informs the server about it via socket

7. **VIEW** - The client displays the information of connected participants.
8. **DELETE** - The client may delete a channel after proper validation and updates the server using HTTP POST request.
9. **MESSAGE** send - The client sends a message to the server including the information of the target user and other metadata. The server checks the validation of the source and destination of the message and forwards the message accordingly. If the broadcast button is on, the message is sent to every channel and user connected to the server. This transfer of messages takes place via socket connection.
10. **DISCONNECT** - When a client leaves, it sends a notification to the server via socket connection. The server then updates the rest of the users about the same.

Languages and Frameworks used:

1. Server

- a. Node-js Language
- b. Nodemon - For ease in debugging
- c. Express-js - Middleware to create the webapp
- d. Socket.io - For socket connections

2. Client

- a. React-js Language
- b. React-Bootstrap - For frontend designing
- c. Socket.io-client - For socket connections

Directory Structure:

chat-app <ul style="list-style-type: none"> ├── api <ul style="list-style-type: none"> ├── package.json ├── package-lock.json └── server.js ├── LICENSE ├── README.md └── ui <ul style="list-style-type: none"> ├── package.json ├── package-lock.json ├── public <ul style="list-style-type: none"> ├── favicon.ico ├── index.html ├── manifest.json └── robots.txt ├── README.md └── src <ul style="list-style-type: none"> ├── App.css ├── App.js ├── index.css ├── index.js └── ToggleSwitch.scss 	Api This directory contains backend server code Server.js Main server code UI This directory contains client code App.js This is the main client code App.css Style sheets for the client
--	---

Sample Snapshots

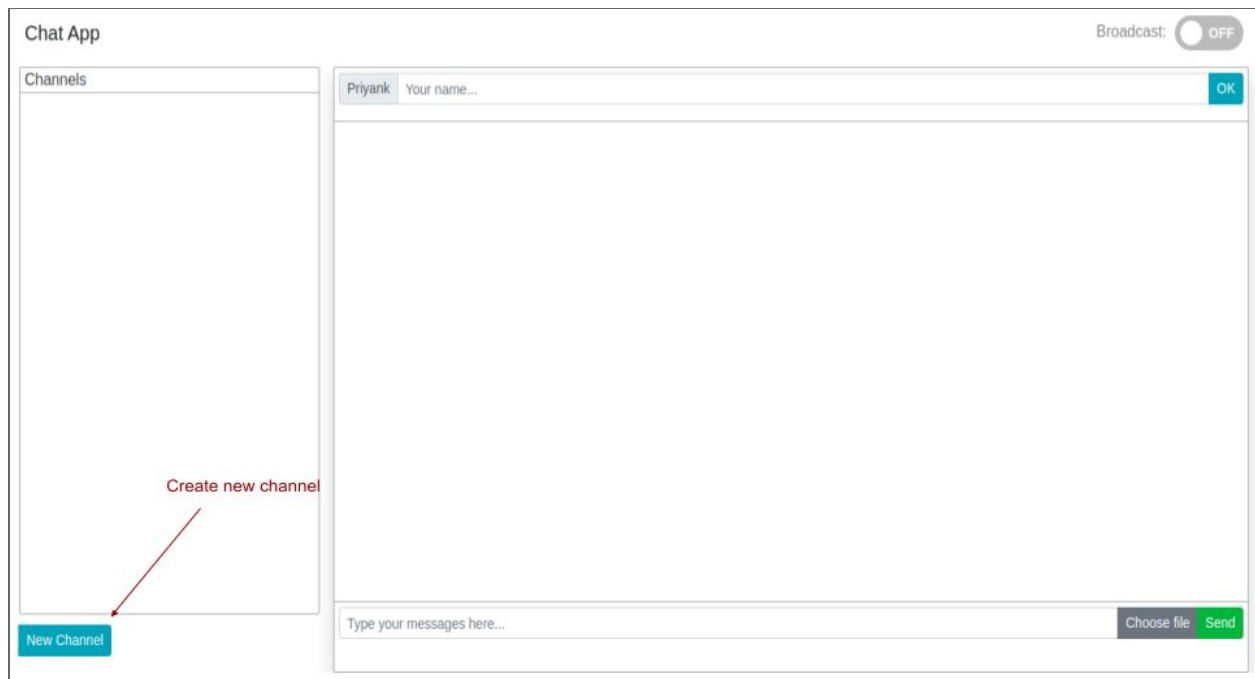


Fig 1. Main page of the webapp

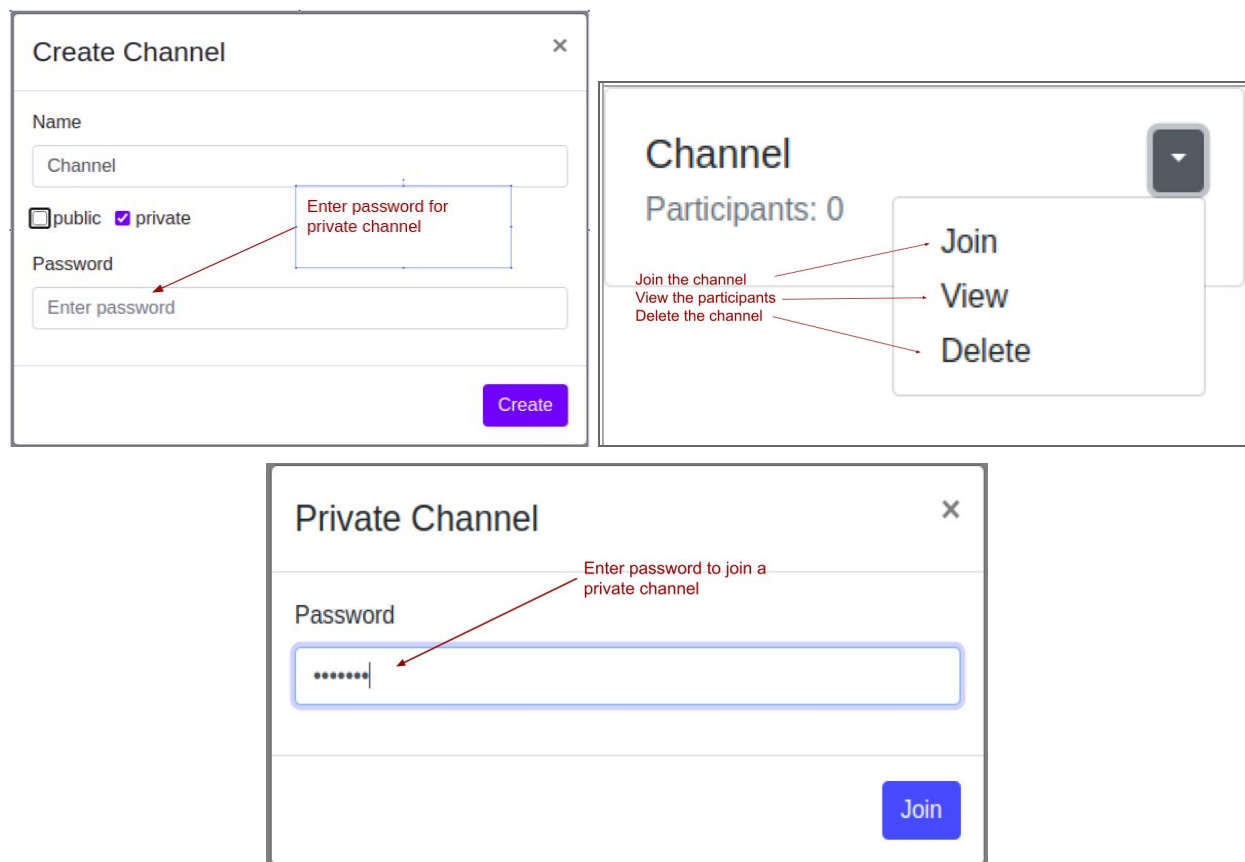


Fig 2. Channel Options

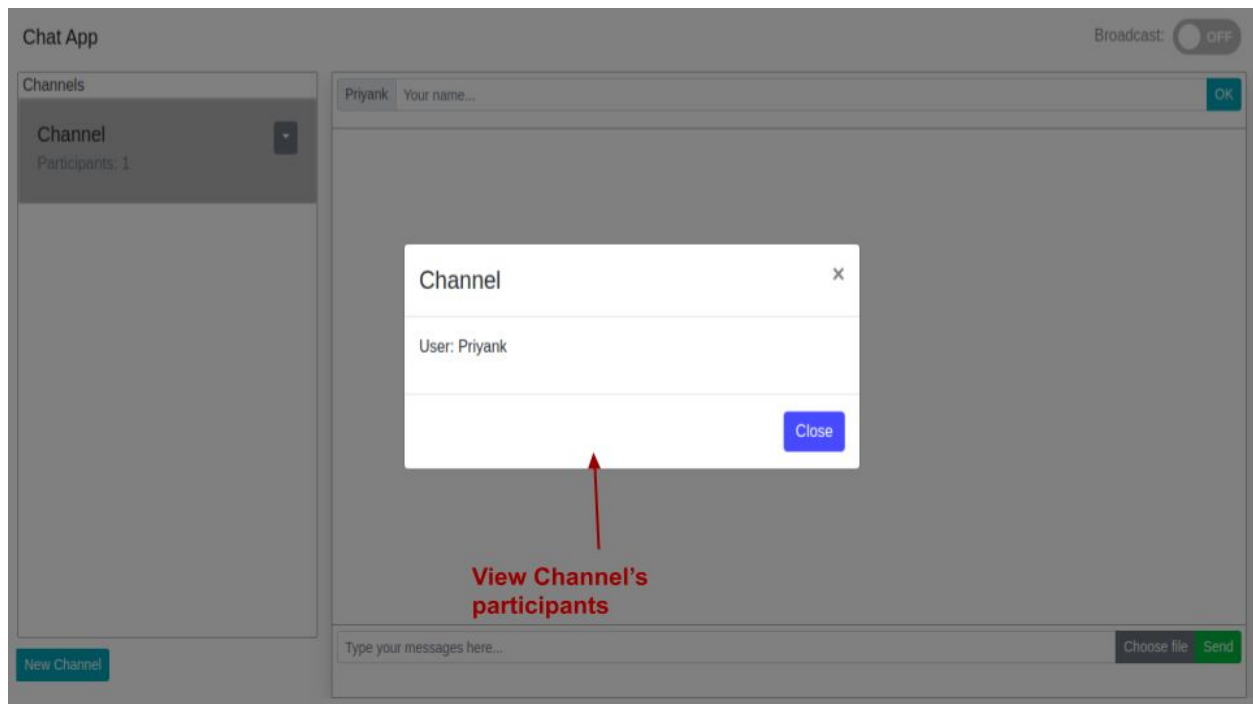


Fig 3. View Channel option showing connected users

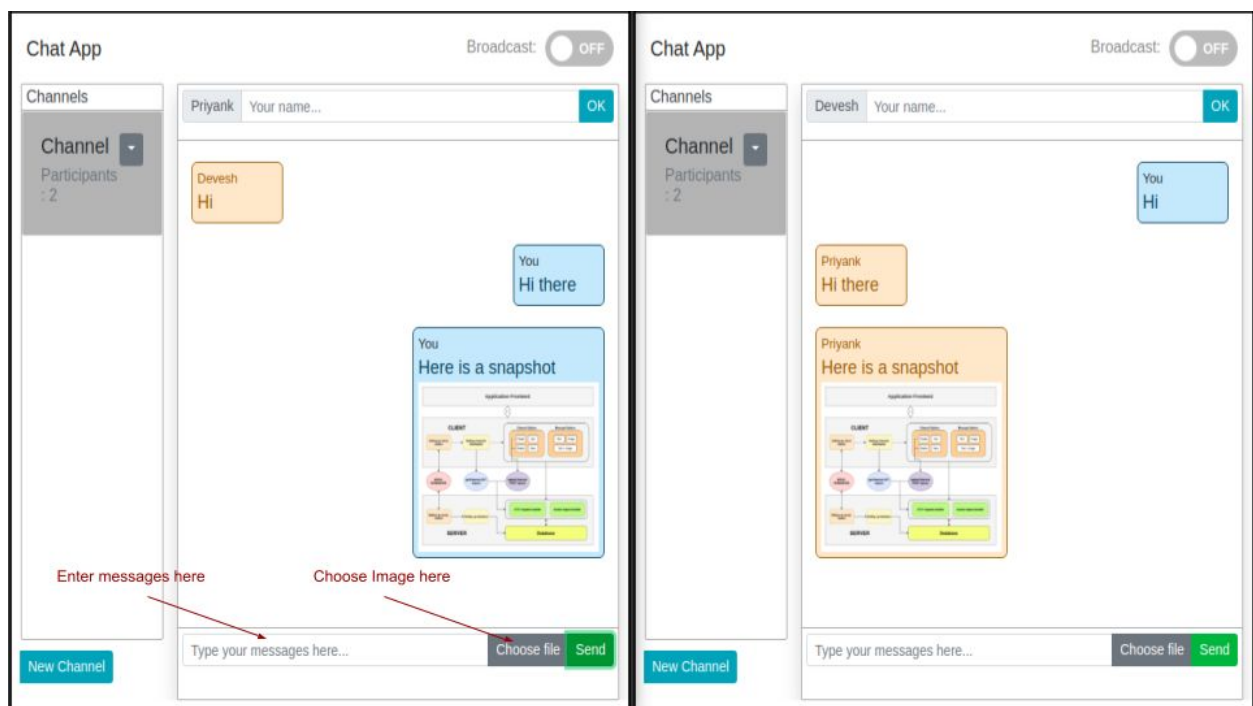


Fig 4. Messaging option view

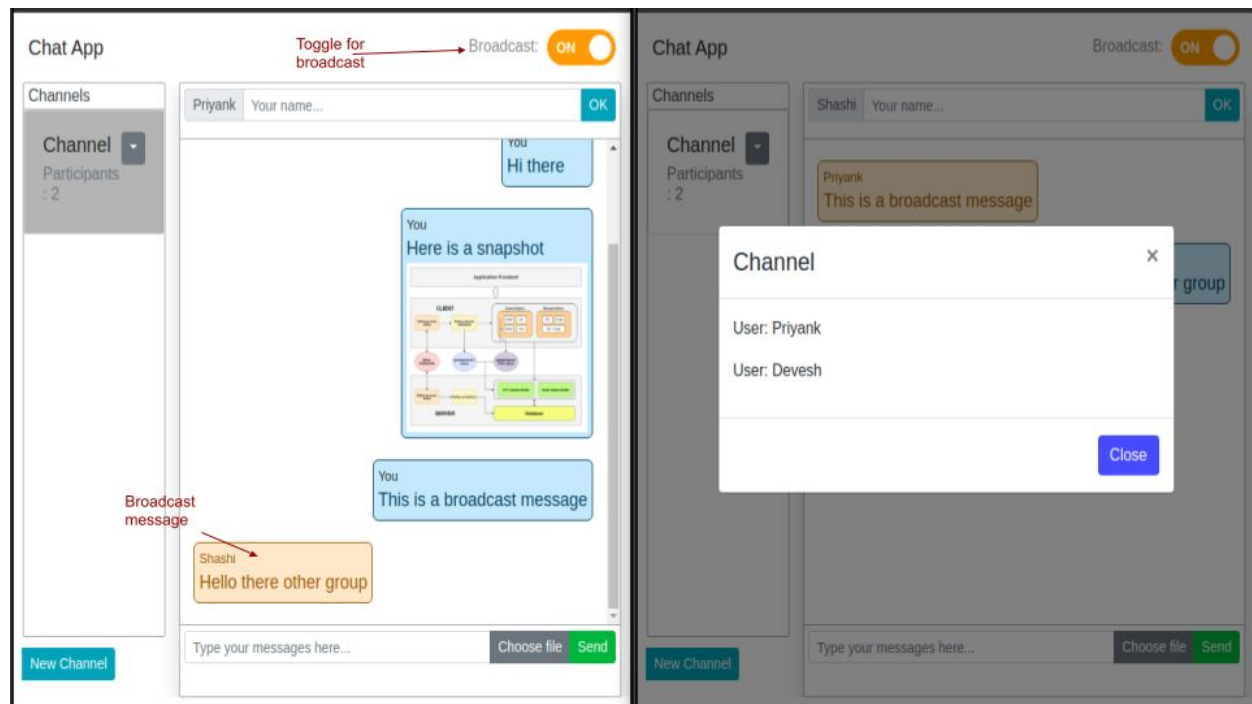


Fig 5. Broadcast message demonstration