Issues:

1. DB not connected error
2. Email path not being received
3. Gender not recognizing ‘male’ as valid enum value

Changes:

1. backend/config/database.js: dotenv.config(); was missing && it appears the IP address keeps changing for some reason, therefore we had keep whitelisting the new address every few times.
2. backend/controllers/userController: await User.create was missing the “email” field, causing it to not be passed when creating user.
3. frontend/src/components/SignUp.js: fixed typo of lowercase ‘m’ to uppercase ‘M’

Redux Store: npm i @reduxjs/toolkit react-redux

Socket implementation:

1. created socket.js inside socket folder in backend
2. inside frontend: npm install socket.io-client
3. After socket changes were made, {app, server} were imported into index.js of backend, and changed app.listen(PORT, ()=>{…} to server.listen…

Deployment:

First issue: Cannot get src/build/index.html

The build was not creating the index.html file because of an issue in one of the CSS files, where I had used irregular characters “<!-- …”. This caused a compilation error that created assets under build/dist folder, but did not contain the other js and html files. It was resolved by removing the characters and correcting the CSS code.

Second issue: Render deployment failed status 127 and 1

We faced an issue where the deployment kept failing on build. Later, it was discovered to be a problem of “nodemon”. So, in package.json file, our start script had included “nodemon backend/index.js”. It was resolved by changing “nodemon” to “node”. Since nodemon is only for development environment and node is used for the deployment, so Render did not recognize it.

Third issue: localhost changed and endpoints

When deploying, we must change localhost endpoints to the suitable endpoint. The “http://localhost:3000/api/v1/user” and such were changed to USER\_API\_ENDPOINT. Then, this variable was declared in frontend/src/utils/constant.js file. Alongside, MSG\_API\_ENDPOINT was also created and replaced the local endpoints.

Fourth issue: backend localhost

Previously, we had been using 8080 as the port for backend. However, as we deploy it, we need to use the respective link to the new URL where we are hosting it. In this case, it was: https://messagingapp-tibe.onrender.com so, we replaced all localhost:8080 with this link, i.e. index.js, useSocket.jsx and App.js.

Fifth issue: DB not connecting

After all the previous changes for remote deployment, the database was not connecting. After scanning various pages, the issue was found to be with the socket.js, which was still connected to localhost:3000 instead of the link to the render url: https://messagingapp-tibe.onrender.com