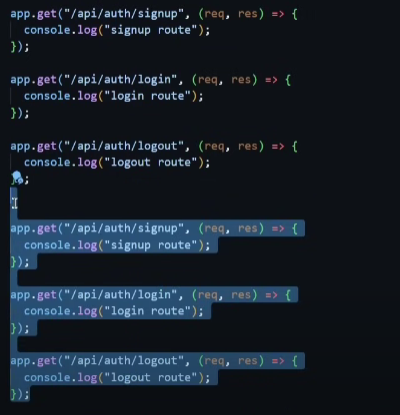
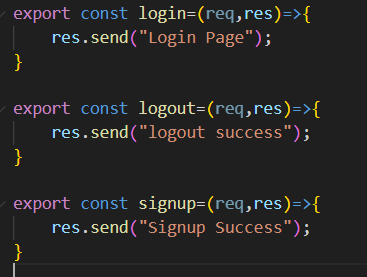
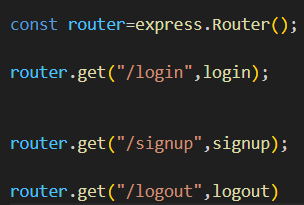
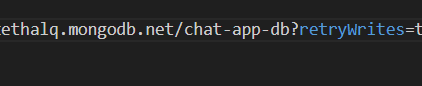
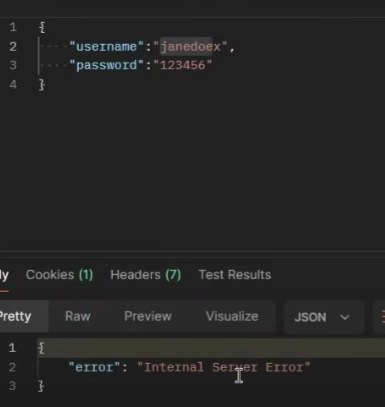
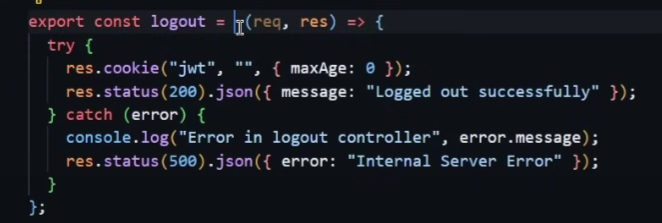
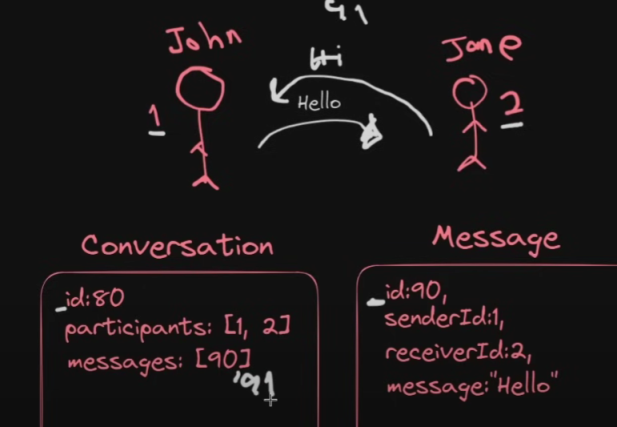
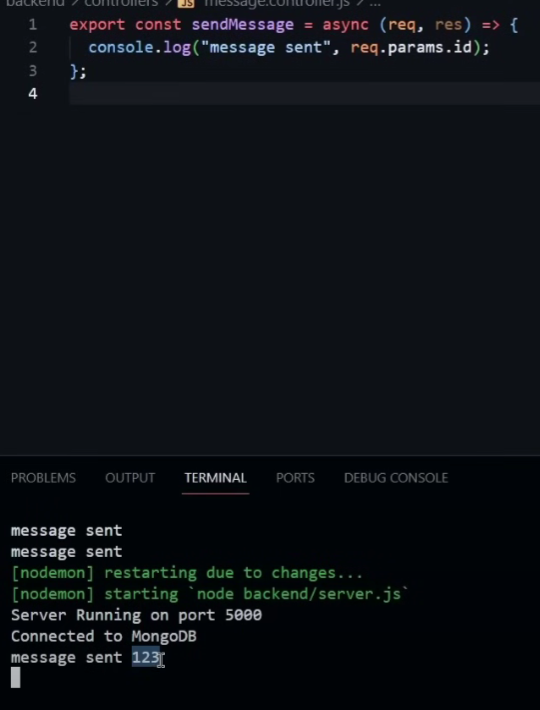
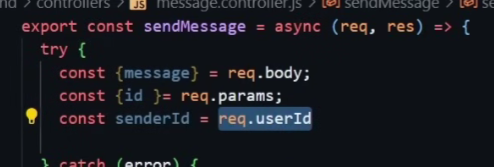
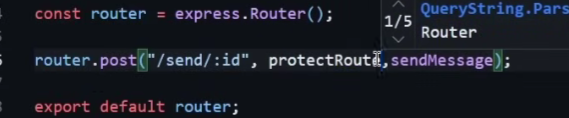
1. What we will use is socket.io for realtime messaging,reactJS for handling UI of our app, nodeJS along with expressJS to handling routing, MongoDB to store data of our app,JWT tokens for authentication purpose of our login page.We will design models and build relationship between them. We will use tailwind css for styling.
2. Dotenv package is used to config environment variables.Cookie parser to parse the cookies.BcryptJS to hash the password. Nodemon is a dev dependency which is dependency used during development process not during production time.
3. 
4. When we use so much routes on our server file it looks ugly so we use middlewares folder which is initialized as app.use(“/api/name”,fileToBeCalled) with this our server file looks clean.
5. Whenever we have too much code in our routes then we use controllers where we write code and export that .
6. Like in controllers we created these function .
7.  
8. Now we can use this in our routes file.
9. IP address 0.0.0.0/0 is accessible from anywhere.
10. 
11. Chat-app-db is our database name.
12. When User does not exist and we are trying to login
13. 
14. 
15. We don’t get cookies in this bcz we destroyed by setting max age as 0 days.
16. 
17. Database design .
18. Participants in conversion contains user in our conversation.
19. Messages id is same in conversation through which we are getting messages.
20. In signup page our details will be saved in user database and after that when we login then these details will be matched and based on that we will be able to login.
21.  Getting ID.
22. 
23. Using req.params.id we get id what we used in our url for calling the route.
24. 
25. For this we will use a middleware which will check whether user is logged in or not.
26. 
27. SO before you run message route it will go to protectRoute which will check if user is logged in or not.
28. In user route we will fetch these users.
29. 