Zync — Issues I Faced and How I Fixed Them

20 issues documented during the development of the Zync chat application.

#	Area	What Happened	How I Fixed It
1	Kafka	When I used the latest Kafka image it kept crashing because the new version needs a different setup without Zookeeper. I did not want to change the whole setup so I just used an older version that works with Zookeeper.	Switched to confluentinc/cp-kafka:7.4.0 and confluentinc/cp-zookeeper:7.4.0 in docker-compose.yml.
2	MongoDB	MongoDB container was restarting every few seconds. The old Docker volume had data from version 6 but I accidentally used the latest image which is version 8. They are not compatible.	Deleted the old volume using docker volume rm and changed the image back to mongo:6 in docker-compose.yml.
3	NGINX / SSL	NGINX would not start because it could not find the SSL certificate files inside the container. The files were on my machine but not mounted into Docker.	Added the certs folder as a volume mount in docker-compose.yml so NGINX can read the certificates.
4	Frontend Container	The frontend Docker container kept restarting in a loop. It had an old nginx config inside that was pointing to a backend called 'backend' which does not exist in our setup.	Updated the frontend nginx.conf to point to backend1:3001 instead. Had to rebuild the image with --no-cache since Docker was using a cached version.
5	SSL Certificates	Running the openssl command in PowerShell was failing because Windows was messing with the path format in the command.	Switched to Git Bash and used double slash at the start of the subject like //CN=localhost. That stopped Windows from converting the path.
6	Backend Startup	All three backend containers were crashing right after starting because they were trying to connect to Kafka before Kafka was actually ready.	Added health checks to Zookeeper and Kafka in docker-compose.yml and made the backends wait for Kafka to be healthy before starting.
7	Docker Networking	Backends inside Docker could not connect to MongoDB, Redis or Kafka because the .env file was using localhost. Inside Docker, services talk to each other using their service names not localhost.	Changed the backend .env for Docker mode to use service names like kafka:9092, mongodb:27017 and redis instead of localhost.
8	Kafka Local Dev	When running the backend locally with npm run start:dev, it was trying to connect to a host called kafka which only exists inside Docker. My local machine has no idea what kafka means.	Changed KAFKA_BROKER to localhost:9092 in the local .env file. Also made Kafka errors non-blocking so even if Kafka fails, messages still get saved to MongoDB.
9	Port Conflict	When I tried to start the backend locally it said port 3001 was already in use. That was because the Docker backend container was already running on the same port.	Stopped the Docker backends first with docker compose stop backend1 backend2 backend3 before running npm run start:dev.
10	Socket.IO	The frontend kept showing a socket connection error. The socket was trying to connect to port 3000 which is the frontend itself instead of the backend on port 3001. This happened because window.location.origin was used as the socket URL.	Changed useSocket.js to use http://localhost:3001 directly as the socket server URL.
11	Messages Not Showing	Messages were being sent from the frontend and the backend was receiving them but they were never showing up in the chat. The backend was getting stuck waiting for Kafka to respond which was timing out so it never got to save the message or send it back.	Moved the Kafka call to run in the background without awaiting it. The message now saves to MongoDB and gets sent back to the clients right away and Kafka happens separately.
12	Dummy Users	Even after building the users module the sidebar was still showing Alice, Bob, Charlie and Diana instead of real users.	Found a hardcoded users array still sitting in Chat.js. Removed it and made sure the initial state was an empty array.
13	Users Not in DB	After building the users module other users could not see my account because my profile was never saved to MongoDB. I had logged in before the save logic was added.	Logged out and logged in again. The login now calls the save API which stores the Google profile in MongoDB.

#	Area	What Happened	How I Fixed It
14	Chat Resetting	Every 10 seconds the selected user was jumping back to the first person in the list and messages were disappearing. The user refresh interval was resetting the selected user every time it ran.	Added a check so the selected user only gets set automatically when no one is selected yet. After that the interval just refreshes the user list without resetting the current chat.
15	Docker Command	The docker-compose command was not found on my machine. Newer versions of Docker Desktop do not use the old docker-compose tool anymore.	Changed all commands from docker-compose to docker compose without the hyphen. This is the new way to run it with Docker Desktop V2.
16	Docker Compose File	Every time I ran docker-compose I was getting a warning about the version field being obsolete in the docker-compose.yml file.	Removed the version line from the top of docker-compose.yml. The new Docker Compose does not need it.
17	Firebase Env	The backend containers kept warning that FIREBASE_PROJECT_ID was not set and defaulting to blank. This was causing auth to not work properly.	Created a .env file in the root chat-app folder with the Firebase project ID so docker-compose could pick it up.
18	Test Files	The Login test file was failing because waitFor was being used inside the tests but was never imported at the top of the file.	Added waitFor to the import from @testing-library/react.
19	Test Files	The Chat test file was failing because the import for the Chat component was written before the jest.mock calls. Jest needs all mocks to be set up before anything is imported.	Moved all jest.mock calls to the top of the file before the import statements. Also cleaned up duplicate mock setup in beforeEach.
20	ESLint Warning	ESLint was throwing a warning about a missing dependency in a useEffect hook in Chat.js. It was asking for fetchUsers to be in the dependency array which would cause it to run too often.	Added an eslint-disable comment on that line to intentionally skip the warning. The hook is meant to only run when the user changes not on every render.