

Spotify clone

Mateusz Mańczak

Technological stack

Frontend

- Next.js / React
- TailwindCSS

Backend

- Node.js
- Express.js
- Prisma ORM

Mixed / Architecture

- TypeScript
- Socket.io
- Docker
- Postgres

Short description

Music player based on websockets where many users can control currently played song at the same time. They may also chat each other with encryption.

Live demo:

<http://10.100.6.171:3000>

Add new song to library

Title

Song title

Audio file

Choose File No file chosen

Cover image

Choose File No file chosen

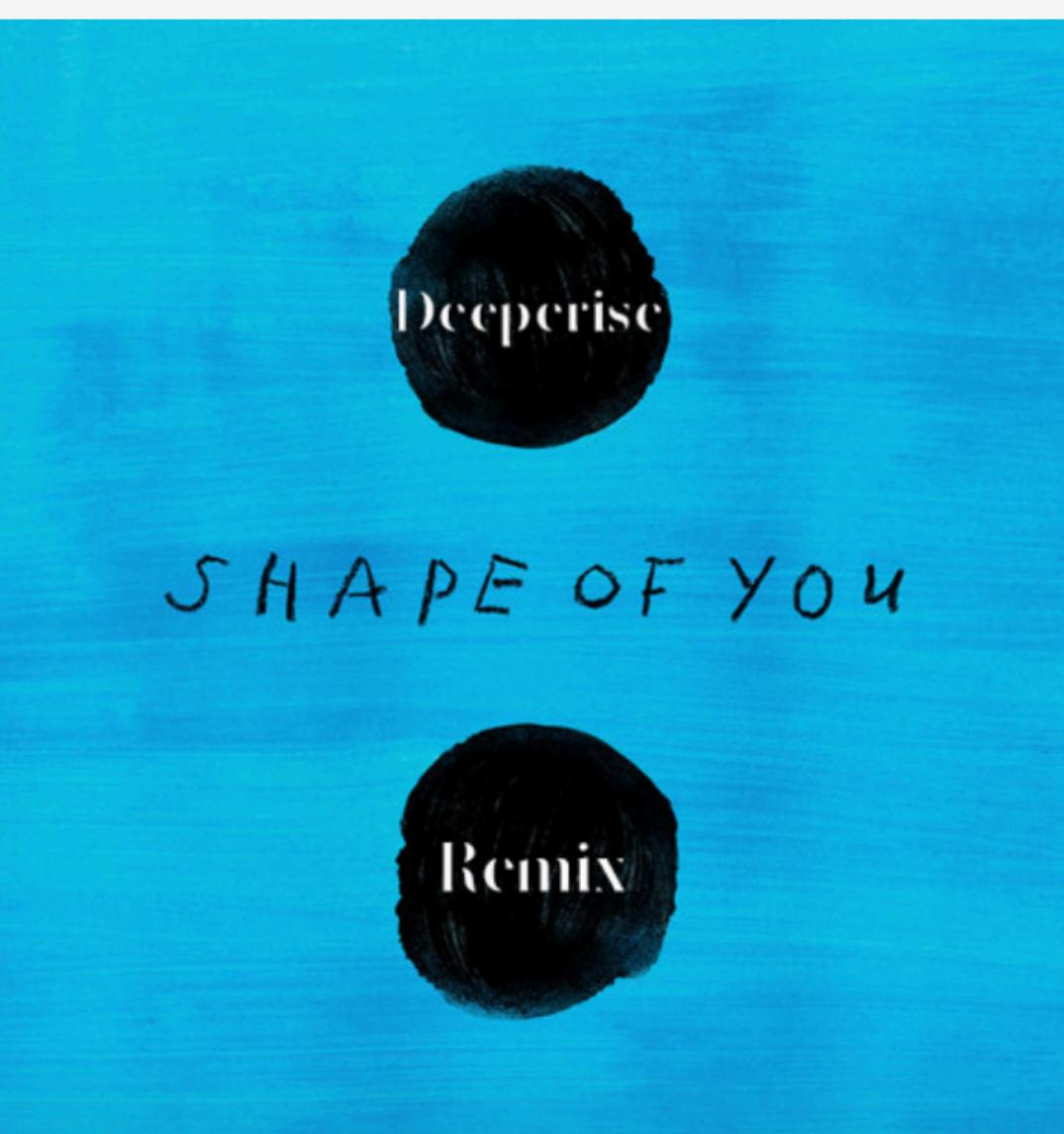
Add to library

Song added to library

Songs



Ed Sheeran - Shape of You



Ed Sheeran - Shape of You



Chat

Shelbi (You)

[SERVER] Unable to decrypt message

Elicia

Unable to decrypt message

[SERVER] Unable to decrypt message

Kirsteni

Unable to decrypt message

Annadiana

Unable to decrypt message

[SERVER] Unable to decrypt message

[SERVER] Unable to decrypt message

Kirsteni

Unable to decrypt message

Add new song

Add new song to library

Title

Song title

Audio file

Choose File No file chosen

Please fill out this field.

Cover image

Choose File No file chosen

Add to library

All fields required

Accepts only mp3 + jpg

File uploading hanled with
“multer” library

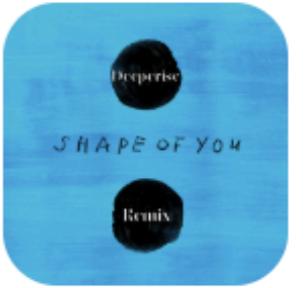
Live updates on others
users' devices

Tracklist

Songs



White 2115 - California

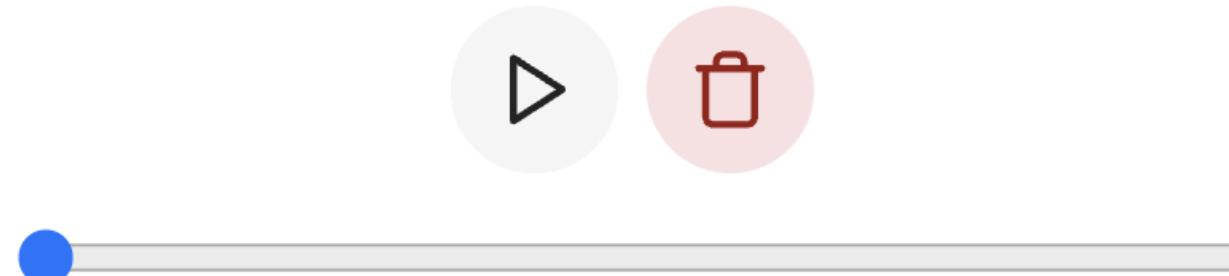
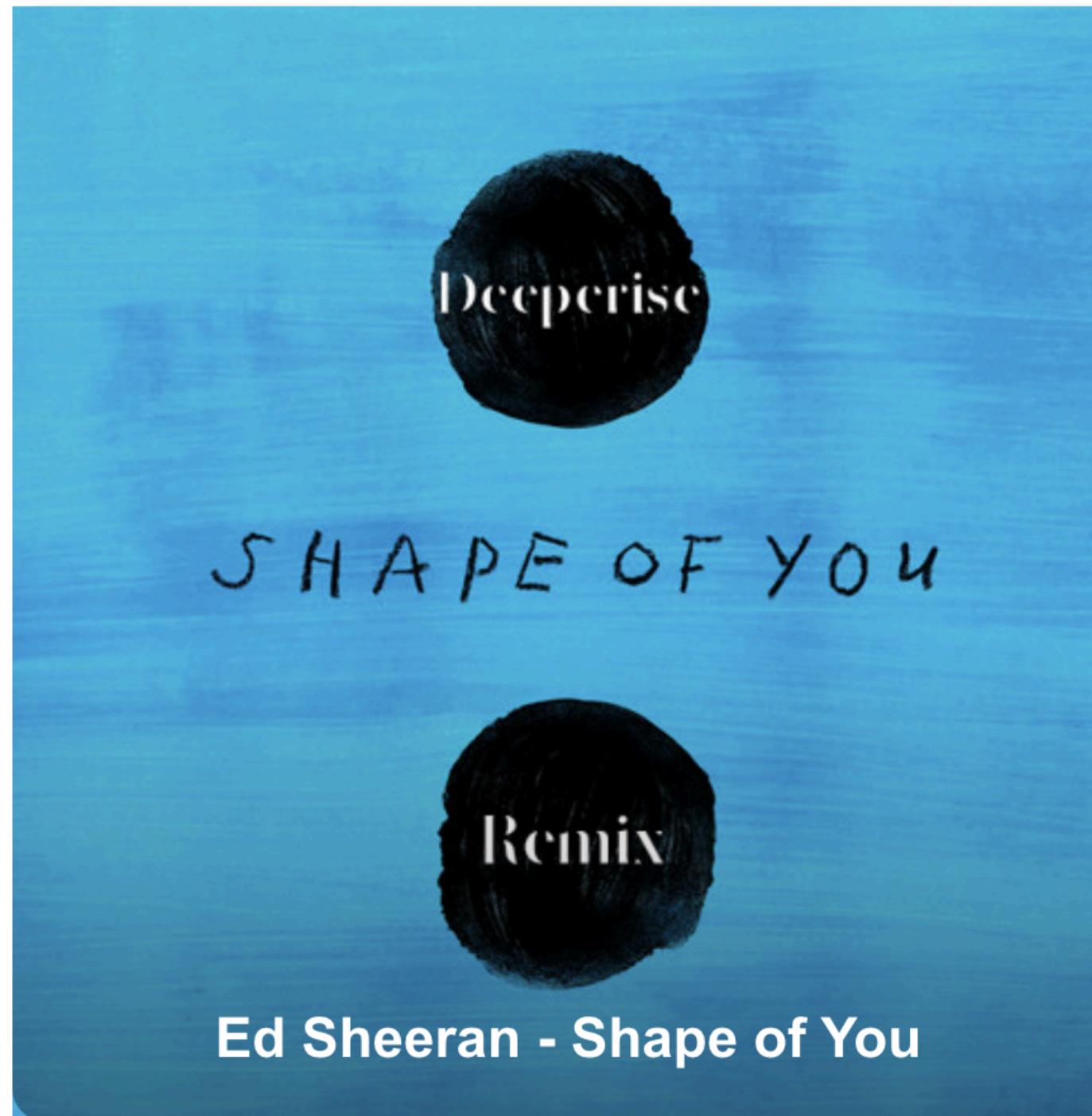


Ed Sheeran - Shape of You



Simple as that

Music Player



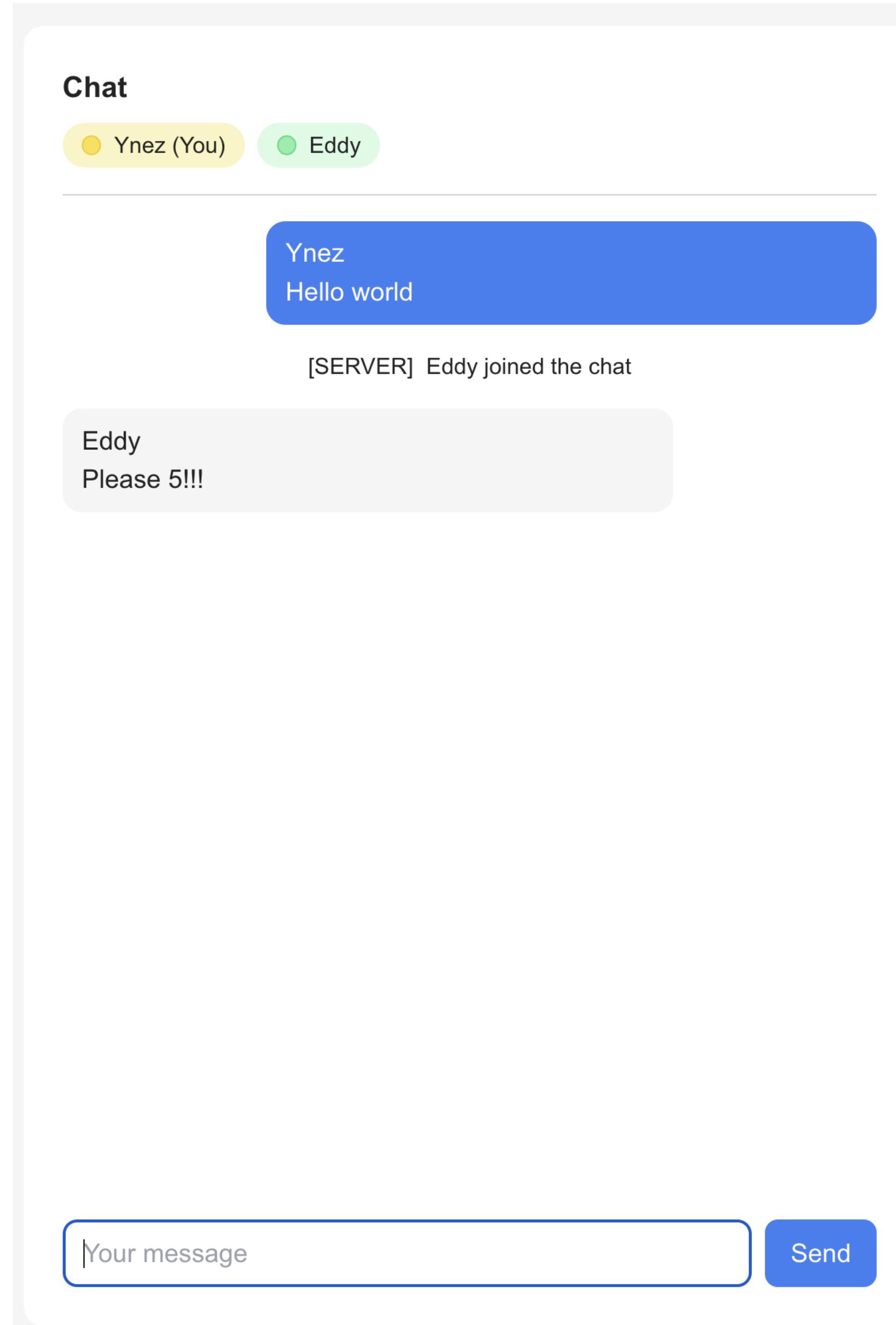
Realtime updates

Streaming audio files

Play/Pause

Change progress

Delete song from library



Chat

Realtime updates

Join/leave notifications

Symmetric encryption
aes-256-cbc

List of users

Active status

Problems during development and solutions

1. UI & Styling
2. Architecture
3. File uploading
4. Realtime communication
5. Encryption

1. UI & Styling

React & Next.js for UI and client state

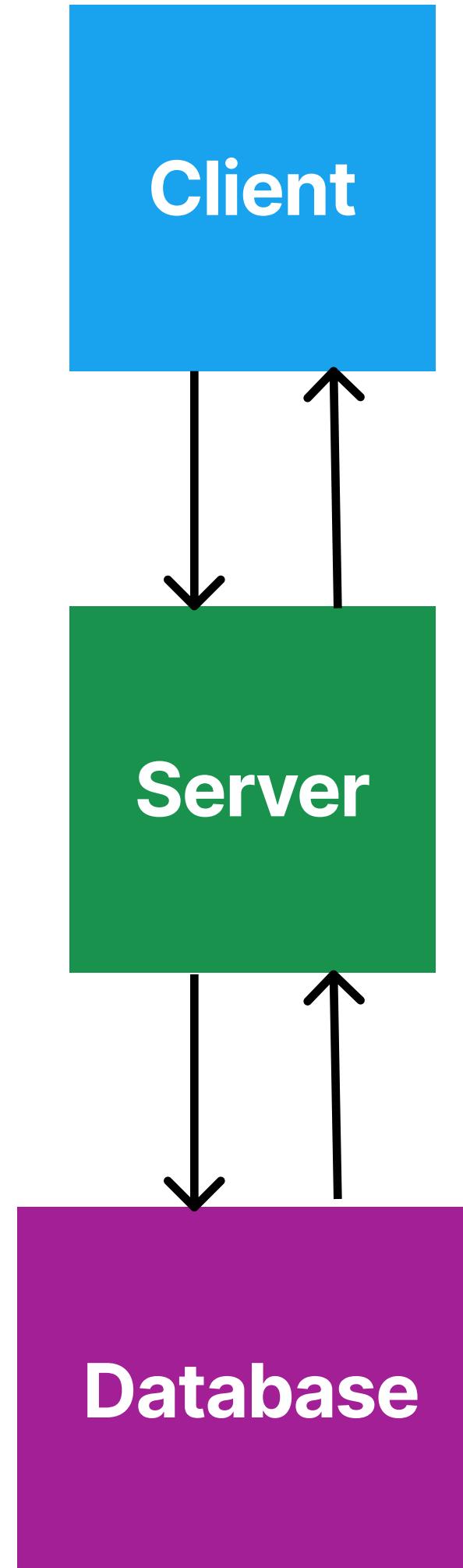
TailwindCSS for styling

2. Architecture

Database: Postgres running on Docker container

Backend: Running on my laptop no not pay for the VPS

Frontend: statically rendered HTML/JS bundle served from my laptop



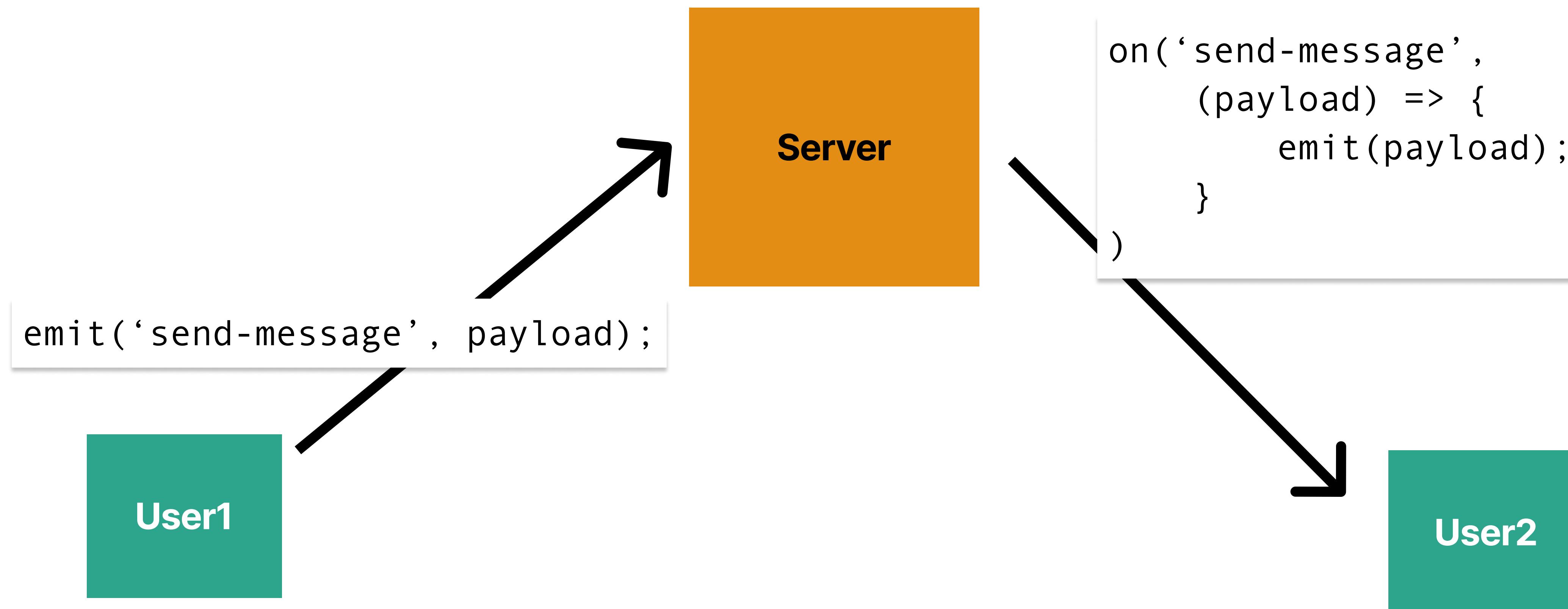
3. File uploading

Sending request from <form> with header
{ “enctype”: “multipart/formdata” }

Backend handles it with “multer” library. Uploads it to the /uploads directory and puts it’s path to the database

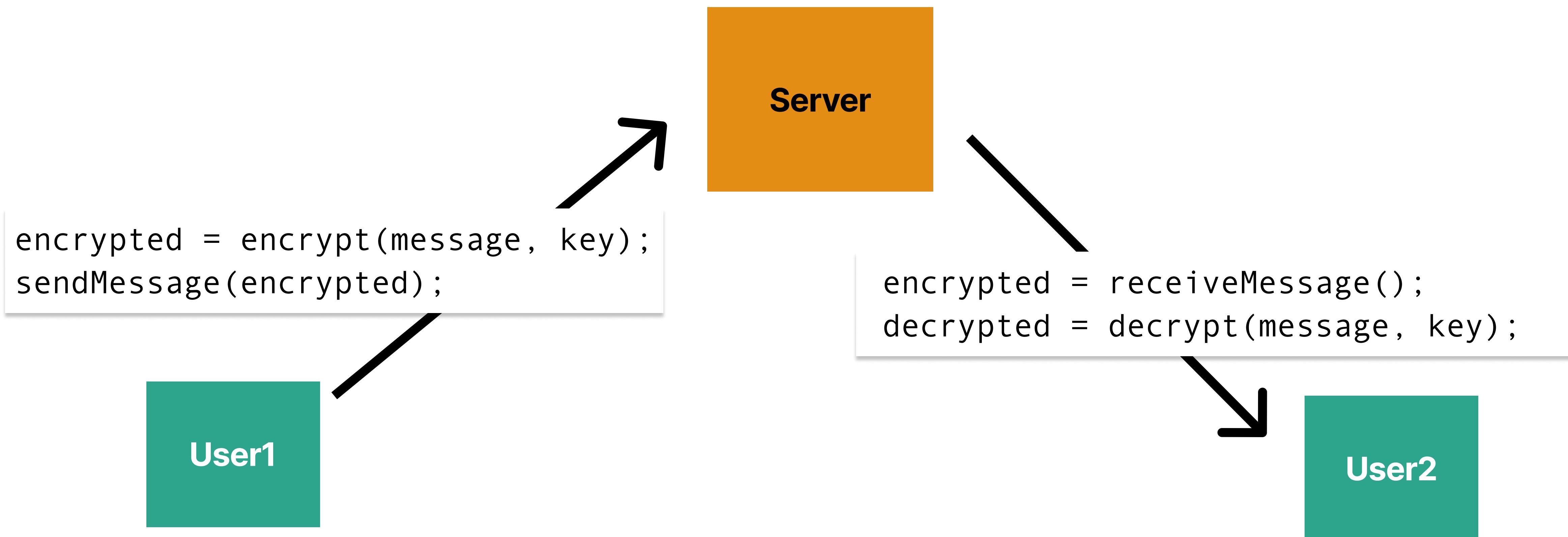
4. Realtime communications

Socket.io



5. Encryption

Symmetric key in memory, for simplicity



Time for demonstration

<http://10.100.6.171:3000>