Starting client and server:
To start the frontend (PORT 3000) : cd client/ & npm run start:frontend
To start the backend (PORT 8000): cd server/ & npm run start:backend

After starting the client, it takes to http://localhost:3000/
It is the homepage where users can create an account or login

For new users, after correctly filling in the email and password, it takes them to the OnBoarding page where user can fill in their personal information . http://localhost:3000/onboarding

For existing users, after loogin in they are taken to the dashboard page where they can chat with their matches and swipe on other users. http://localhost:3000/dashboard

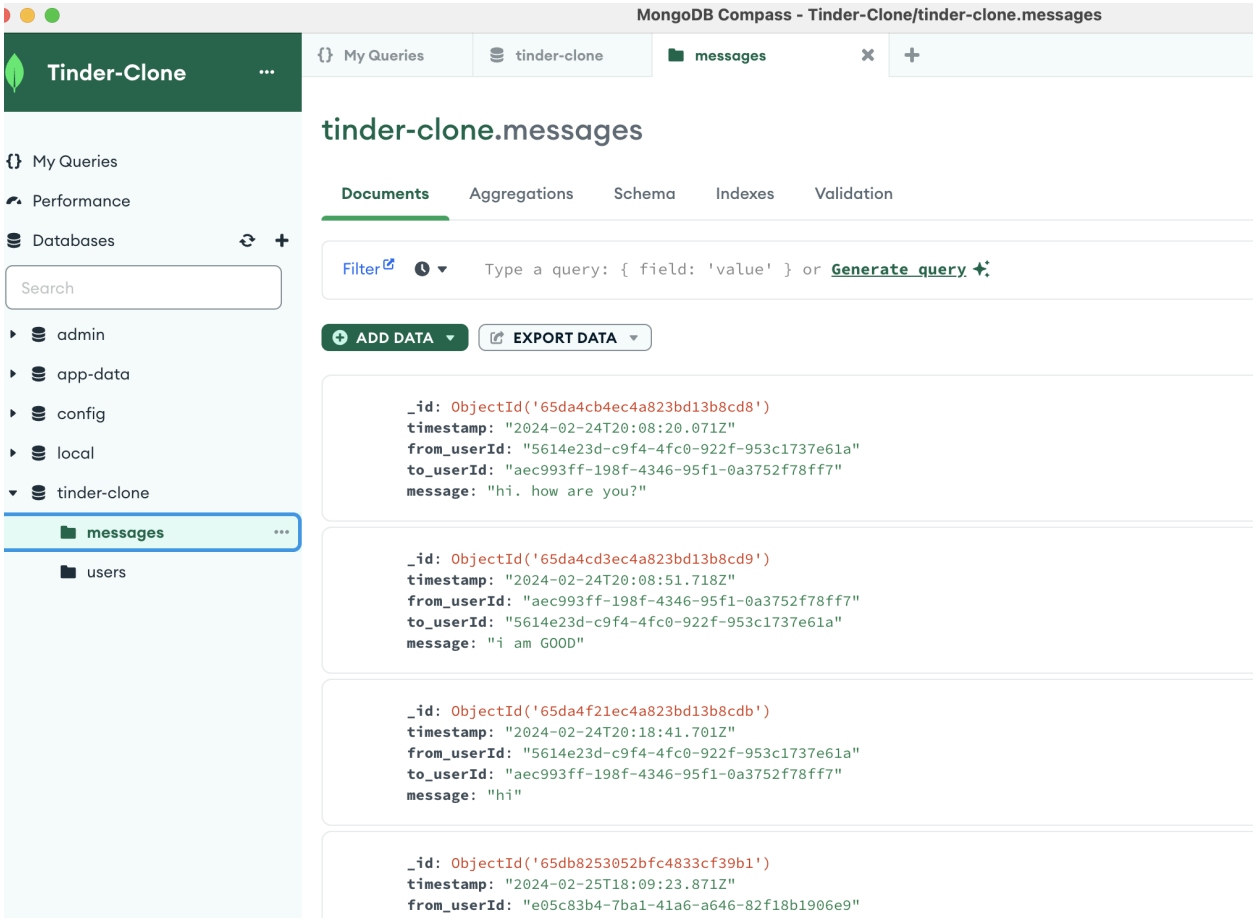
Swiping features:
Note: Theses are the dependencies and version for react-tinder-cards: `"react-tinder-card": "^1.5.4", and "@react-spring/web": "^9.5.5",`

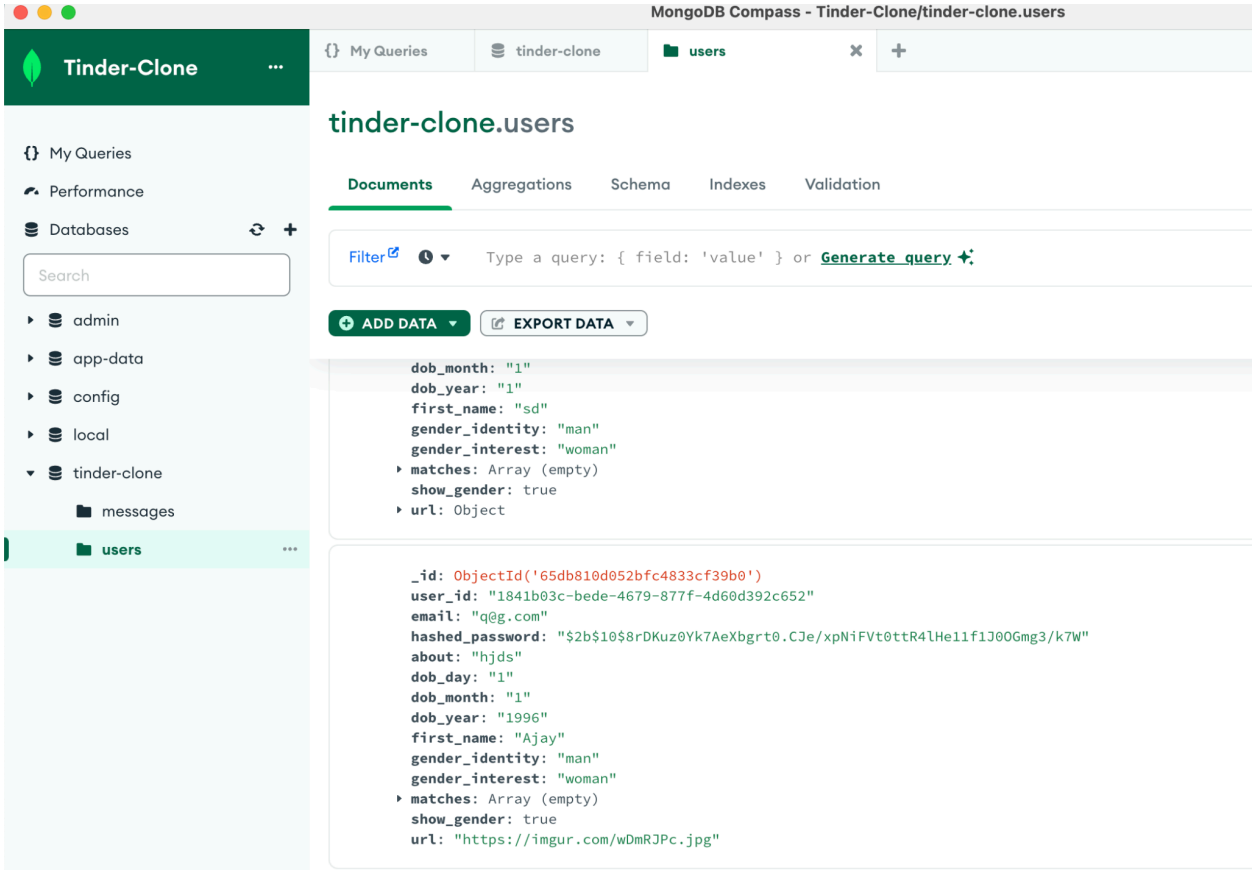
I used them for adding swiping features.

Technology choices and Installation:

1. React and Node.js express

2. I used MongoDB for storing the users and their messages in the database. Here is the mongodb link: `mongodb://localhost:27017/tinder-clone`


MongoDB Compass - Tinder-Clone/tinder-clone.messages
Tinder-Clone
My Queries
Performance
Databases
Search
admin
app-data
config
local
tinder-clone
messages
users
My Queries
tinder-clone
messages
tinder-clone.messages
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value' } or Generate query
ADD DATA
EXPORT DATA
_id: ObjectId('65da4cb4ec4a823bd13b8cd8')
timestamp: "2024-02-24T20:08:20.071Z"
from_userId: "5614e23d-c9f4-4fc0-922f-953c1737e61a"
to_userId: "aec993ff-198f-4346-95f1-0a3752f78ff7"
message: "hi. how are you?"
_id: ObjectId('65da4cd3ec4a823bd13b8cd9')
timestamp: "2024-02-24T20:08:51.718Z"
from_userId: "aec993ff-198f-4346-95f1-0a3752f78ff7"
to_userId: "5614e23d-c9f4-4fc0-922f-953c1737e61a"
message: "i am GOOD"
_id: ObjectId('65da4f21ec4a823bd13b8cdb')
timestamp: "2024-02-24T20:18:41.701Z"
from_userId: "5614e23d-c9f4-4fc0-922f-953c1737e61a"
to_userId: "aec993ff-198f-4346-95f1-0a3752f78ff7"
message: "hi"
_id: ObjectId('65db8253052bfc4833cf39b1')
timestamp: "2024-02-25T18:09:23.871Z"
from_userId: "e05c83b4-7ba1-41a6-a646-82f18b1906e9"



3. I use uuid for generating unique user IDs for each user.

4. I also used CORS middleware to enable HTTP requests from backend to frontend.
5. I used bcrypt for hashing passwords and jsonwebtoken for Authentication and authorization

Additional features:

Besides the basic feature implementation which gives 25 points, I have following features:
Utilization of a frontside framework, such as React - 5 points
One can swipe to left or right to dislike or like the profile- 2 points
If match is being found the UI gives option to start chat immediately - 2 points
User profiles can have images which are shown on the main page and in the chat - 3 points

Additional features of my own:

1. Have created an OnBoarding page where users can fill in their informations like name, DOB, write their bio
2. I have also set images to be required so that it looks better for swiping
3. I, as an admin, I can view who matched with in the MOngoDB database.I can also generate the JSON list of all users using: http://localhost:8000/listOfUsers
4. Users can set their preferences whether they like man or woman. They can choose gender_interest on the OnBoarding page and then dashboard page will show them man/woman based on their preferences.

For images URL:
I have used https://imgur.com/

User Accounts (Demo Purpose):
Females:

1. scarlett@gmail.com
2. emily@gmail.com
3. emma@gmail.com
4. blake@gmail.com
5. keira@gmail.com

Male:

1. tom@gmail.com
2. will@gmail.com
3. james@gmail.com
4. hanks@gmail.com
5. bob@gmail.com

Format for uploading the image: https://imgur.com/h74hvSz.jpg

And the password is 123 for testing purposes

Source: I followed one tutorial video for this project. Here is the link:
https://www.youtube.com/watch?v=Q70IMS-Qnjk