

EMOTIONS

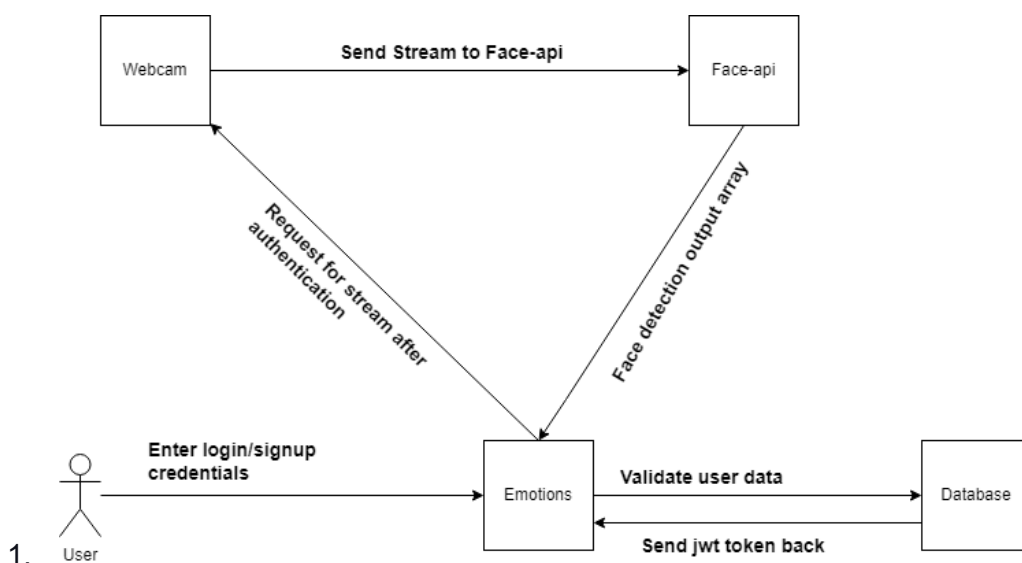
for lambesideyou. inc

The purpose of this document is to serve as a quick overview of the website Emotion I created for lambesideyou. The document will walk through everything I considered while designing the website and how I tried to implement it.

Emotion is a website where we use AI to detect a person's emotion in real-time. For this, Emotions have used a modern library, FaceApi.js, and the website's structure is in the MERN stack. FaceApi.js provides various functionalities, but I specifically used emotion detection and face landmark functionalities.

As website require real-time detection, I have integrated AI in the frontend to decrease latency, and also, it does not require sending a request to the backend every 100ms.

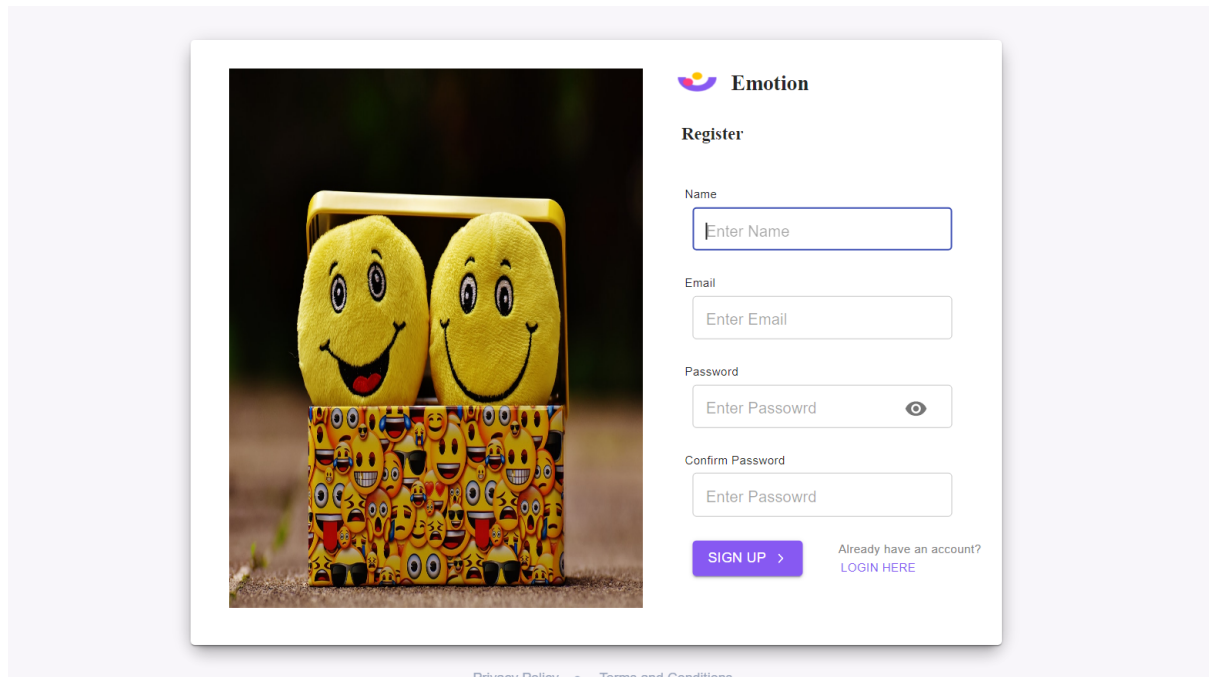
The document will go through the front end and backend separately. Emotion has used **git** for version control, and **npm** is the package manager.



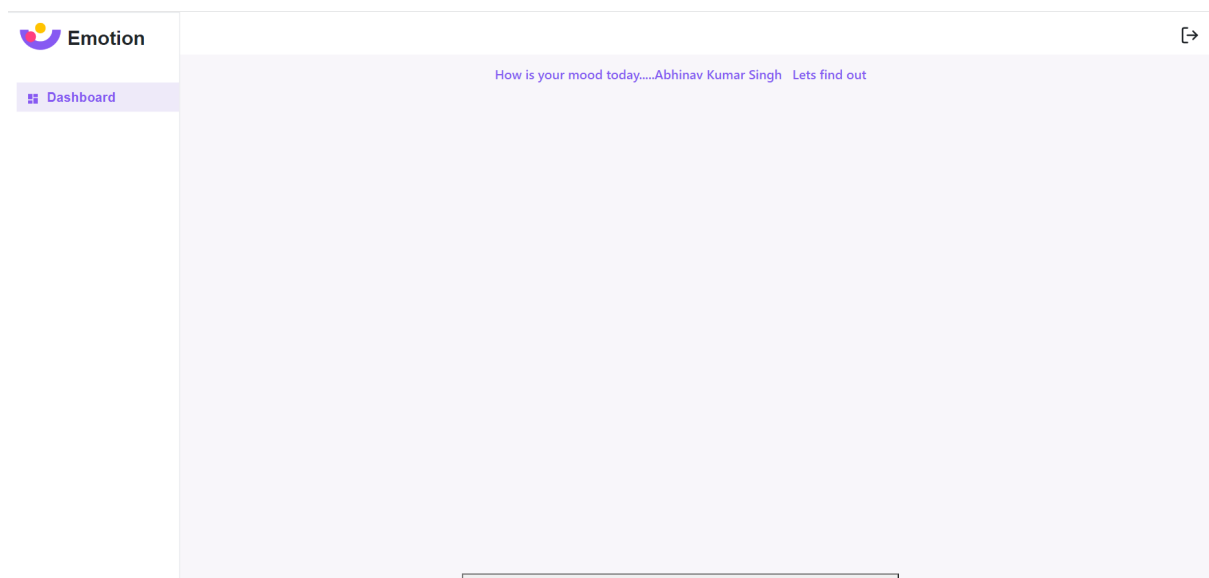
Frontend

The frontend is created with create-react-app, which takes care of all configurations. Emotions have two pages, one for login one for emotion detection.

1. I have placed all pages in the components folder and reusable pieces of code.
2. To send requests, Axios is being used, which is a promise-based service
3. All Axios configuration is in the API folder in the index file



4. Emotions have used Material UI, which eliminates the need for a separate CSS file
5. Emotions have integrated react-icons to make UI better, and icons are straightforward to use
6. Functional components are made with react-hooks that follow the modern standards for React
7. Models are placed in public folder form where package face-API get all the models through request on their specific URL



- 8 I have not saved any data on the frontend; instead, I have saved jwt and _id in local storage from which user data will verify, and we get back user data on load through use effect

9. Every component that can be reusable is made into a different component

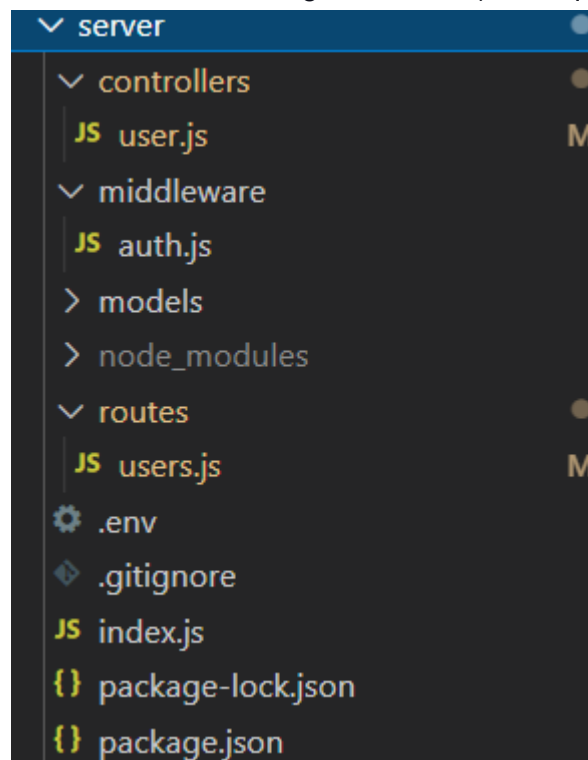
I hope you like the UI:)

Backend

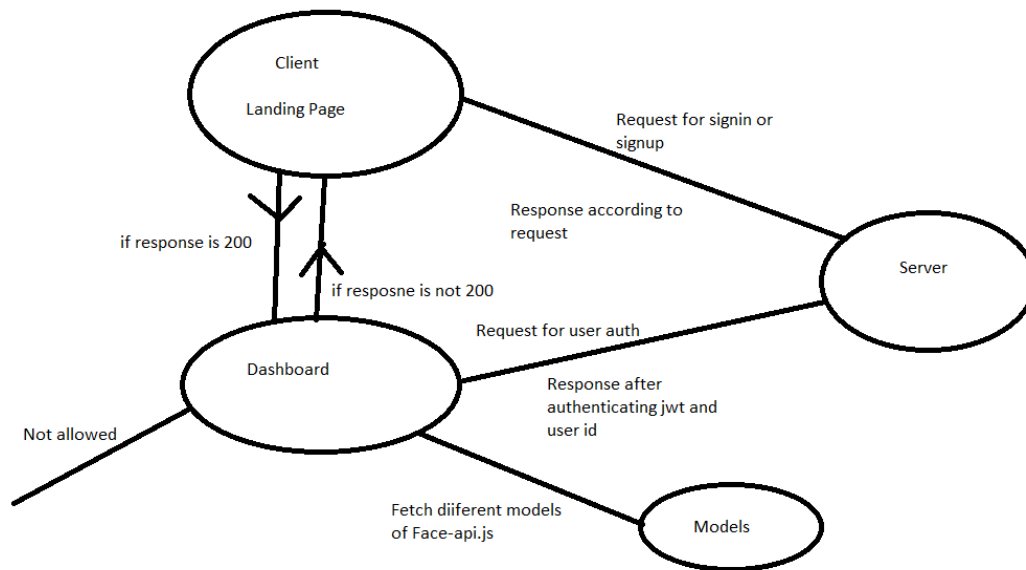
1. Backend is running in Node.js environment with Express
2. To store data, the NoSQL database MongoDB is being used.

As the complexity of the database and queries is not much, there is not much difference in performance for SQL and NoSQL databases, so I have chosen MongoDB according to my preference.

3. For user Authentication, Emotions uses JSON Web Token(JWT)
4. To connect to MongoDB , a connection URL must be specified in .env file
5. There is a single route for Users through which various endpoints are present
 - For signup-"/user/signUp"(Post request)
 - For sign in-"/user/sign" (Post request)
 - For authentication of User and to get user data(Get request)

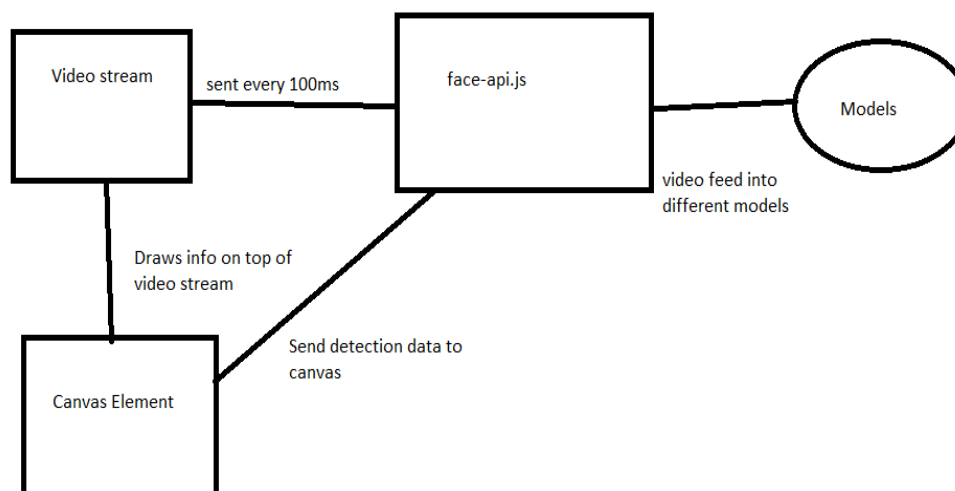


6. Middleware auth.js authenticate token in the header of request added by Axios interceptor in the frontend.
7. Controllers have all the logic in the form of functions which makes them very reusable
8. Routes have all different routes for specific types of request



Interaction between Backend and Frontend is limited to user data, as shown in the diagram.

As mentioned, AI functionality is on the front end. Face-api.js takes the video stream from the users' camera through **the navigator.media devices.getUserMedia** and processes it every 100ms. It returns an array with different types of data I generated through video processing, which fed to canvas element that draws these findings on top of the video stream in UI.



It shows the following things.

- Facial features
- Emotion with a level of confidence has
- Overall face position

Conclusion

My idea of making this website is that people often want to ask people if they are looking sad but could not ask because of embarrassment, but now they see their emotion right now with complete privacy. This will decrease anxiety in people.

In last, I would like to thank lambesideyou for giving me this excellent opportunity to learn. I would never think to integrate AI into a website, and that it is quite a fun