

# **너의 머리 속이 보여 (Show your emotion)**

**- 안면 인식 기반 챗봇 웹 서비스**

**최필수**

# 프로젝트 소개

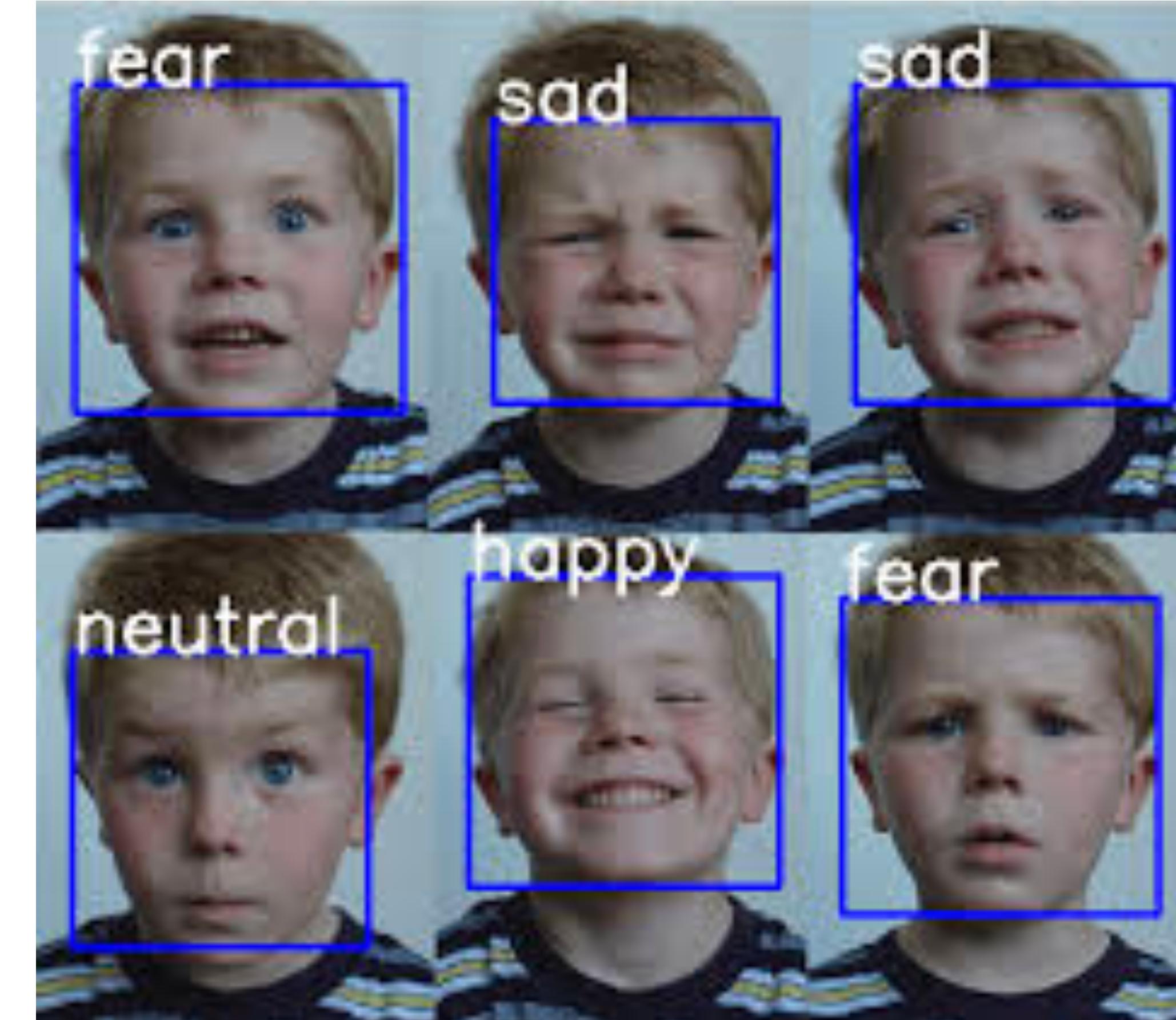
- 프로젝트명 - 너의 머리 속이 보여

주요 기능 :

사용자 안면 인식을 통해

감정을 추출하고

어울리는 조언을 받을 수 있음



- 이미지 출처 : <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSssCYYJL0MFDGx8XTGhwipKQoduednXba4kw&usqp=CAU>

# 개발 동기

- 안면 인식 기술을 활용한 플랫폼 서비스를 상용 수준에서 운영해보고 싶었음
- 최근 다양한 기업들에서 chatbot 시스템을 도입하고 있는데 이를 구현하고

안면 인식과 연동해보고 싶었음

M NEWS 문화일보 | 9면 TOP | 1일 전 | 네이버뉴스

## 학부모 단순·반복민원, AI 챗봇이 처리한다

교육부는 나이스(교육행정정보시스템)에 온라인 민원 시스템을 도입하고 이에 탑재할 수 있는 민원처리 인공지능(AI) 챗봇을 시도교육청과 공동으로 개발해 단순·반복 민원의 부담을 덜겠...

www.aitimes.com › news

## 아마존, 상품 검색에 AI 챗봇 도입한다 < AI 기업 < 산업 < 기사본문 - AI타임스

2023.05.16. 세계 최대 전자상거래 업체인 아마존이 자체 상품 검색 시스템에 인공지능(AI) 챗봇을 도입한다. 블룸버그는 15일(현지시간) 아마존이 최근 선임 엔지니어 채용공고에서 "사용자가 질문할 수 있고 제품을 비교하며 개인화된 제안을 받을 수 있도록 대화형 경험으로 아마존 검...

신아일보 | 3일 전

## SKT, AI고객센터 '페르소나AI' 투자...콜봇·챗봇 추진

음성엔진, 음성인식, 문장분석 등 각종 AI 기술을 적용해 AI챗봇, 콜봇을 통해 24시간 상담이 가능하다.... 은행·보험·증권 등 금융회사 수요가 가장 많고 유통, 레저, 교육, 제조 업체들도 AIC...

한국경제 PiCK | A13면 1단 | 3일 전 | 네이버뉴스

## "기업의 3분의 2가 도입할 것"...통신사들 뛰어든 '새 먹거리'

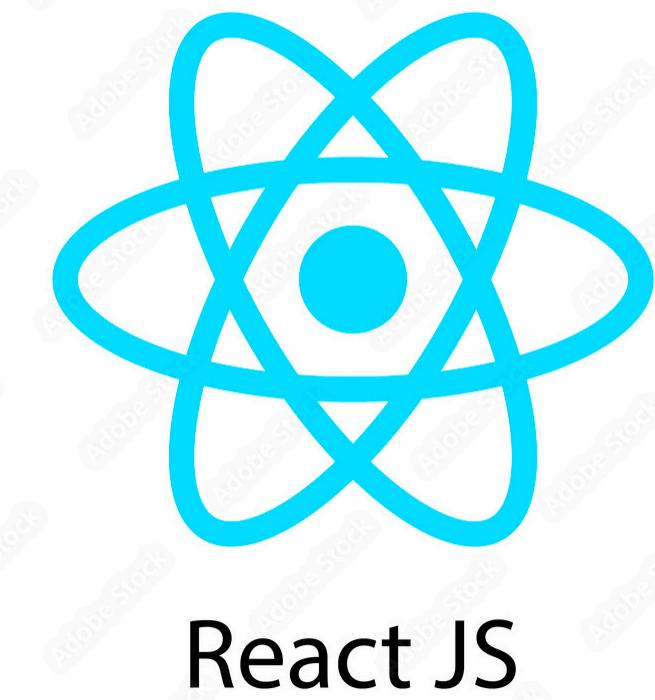
단순 예약, 안내, 고장 접수 업무는 챗봇·콜봇 등 AI가 처리한다. 상담원은 보다 복잡한 업무를... 국내 기업의 약 3분의 2가 AICC를 도입할 것으로 예상한다"고 말했다. 중소기업, 소상공인을...

# 개발 일정

Week & Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14
주제 조사 및 기술 스택 선정														
사용자 UI 개발														
서버 개발														
프론트 서버 연동														
테스트														
배포														
보고서 작성														
제출														

# 기술 스택

- 1. Front -> React
- 2. Backend ( WAS ) -> Django
- 3. Web Server -> Nginx
- 4. Deploy -> Docker, EC2(AWS)



# 웹 페이지 구성

- 회원가입
- 로그인
- 안면 감정 인식 페이지
- 채팅 페이지
- 게시판
- 마이페이지

# 웹 페이지 동작 원리 - 회원가입

- Input Parameter
    - Username, email, password, confirm password
- 프론트엔드단에서 총 4가지 값을 입력 받아 백엔드단으로 전달합니다.
- 인풋 값을 입력받을 때 하단의 요구사항에 맞는 유효성을 검사합니다.
- Username이 3~16 글자 수 이내여야하며 특수문자 불포함 여부
  - 올바른 email 주소 여부
  - password가 최소 1개의 대문자, 1개의 숫자, 1개의 특수문자를 포함하는지 여부
  - confirm password와 password가 일치하는지 여부

# 웹 페이지 동작 원리 - 로그인

- Input Parameter

- Username, email, password

client단에서 총 3가지 값을 입력 받아 백엔드단으로 전달합니다.

로그인 성공 이후 서버단에서 JWT 토큰을 발급하고 client에서는 브라우저 내의 local storage에 저장합니다.

JWT는 채팅 페이지, 감정 인식 기능 등을 이용할 때  
인증 정보로써 활용됩니다.

# 웹 페이지 동작 원리 - 안면 감정 인식

- 안면 감정 인식 활용을 위해선 pytorch 혹은 tensor flow 등을 이용하여 AI 모델을 구축하고 데이터셋을 수집하여 모델을 학습시켜야 하지만 오픈 소스인 face-api.js 모듈은 이미 학습된 모델을 가지고 api를 제공하고 있습니다.

따라서 위의 face-api와 WebCam 모듈을 활용합니다.

노트북 내의 Webcam을 통하여 사용자의 안면을 실시간으로 감지하고 감정을 추출합니다.

# 웹 페이지 동작 원리 - 채팅 페이지

- Open AI 사의 chat-gpt 3.5 turbo 모델의 API를 활용하여 사용자와 gpt가 상호 작용할 수 있는 채팅 페이지를 구현합니다.
- 안면 감정 인식을 통해 감정을 추출하고 추출된 감정을 바탕으로 프롬프트를 알고리즘으로 랜덤 생성하여 gpt에게 전달하고 감정에 대한 조언을 받을 수 있습니다.

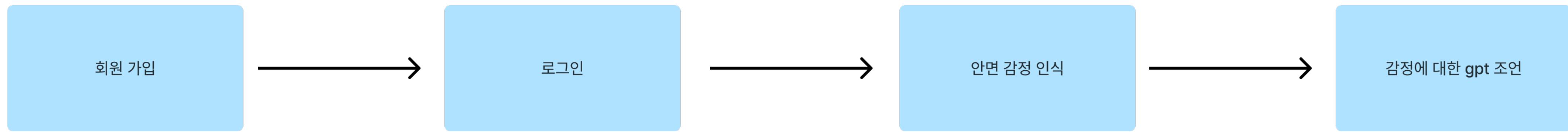
# 웹 페이지 동작 원리 - 게시판

- 여러 사용자들이 일상을 공유할 수 있는 커뮤니티용 게시판을 구현합니다.

# 웹 페이지 동작 원리 - 마이페이지

- 사용자들이 회원 가입시 자신이 등록한 정보를 수정하거나 회원 탈퇴를 할 수 있는 기능을 구현합니다.
- 자신이 등록한 게시글을 확인할 수 있는 기능을 제공합니다.

# 소프트웨어 순서도



# 구현 모습 - 회원가입 & 로그인

Home Chat Board User

## Register

Username

Username

Email

Email

Password

Password

Confirm Password

Confirm Password

Submit

Home Chat Board User

## Login

Username

Username

Email

Email

이 입력란을 작성하세요.

Password

Password

Sign in

Sign up

# 구현 설명 - 회원가입 & 로그인 페이지

```
const AuthPage = () => {
  const initialValue = {
    username: '',
    email: '',
    password: '',
    confirmPassword: '',
  };

  const [values, setValues] = useState(initialValue);
  const [registerMode, setRegisterMode] = useState(false);
  const [auth, setAuth] = useRecoilState(isAuthenticated);
  const [accessToken, setAccessToken] = useRecoilState(authToken);
  let navigate = useNavigate();

  const handleRegister = async (e) => {
    const registerParams = {
      username: values.username,
      email: values.email,
      password1: values.password,
      password2: values.confirmPassword,
    };
    e.preventDefault();
    const registerRequest = await simpleFetch(
      urls.registerPath,
      "POST",
      JSON.stringify(registerParams),
      ""
    );
    e.preventDefault();
    registerRequest();
    setRegisterMode(false);
    setValues(initialValue);
  };
}
```

```
const handleLogin = async (e) => {
  e.preventDefault();

  const loginParams = {
    username: values.username,
    email: values.email,
    password: values.password,
  };

  try {
    const loginRequest = await simpleFetch(
      urls.loginPath,
      "POST",
      JSON.stringify(loginParams),
      ""
    );
    const { access, refresh, user } = loginRequest;
    localStorage.setItem("accessToken", access);
    localStorage.setItem("refreshToken", refresh);
    localStorage.setItem("userInfo", JSON.stringify(user));
    setAccessToken(access);
    setAuth(true);

    console.log("login success");
    return navigate("/chat");
  } catch (e) {
    console.error(e);
    alert("로그인 도중 에러가 발생하였습니다");
    setAccessToken("");
  }
};
```

Register : React 컴포넌트를 활용하여 유저로부터 전달받은 회원 정보를 서버에 전송합니다.

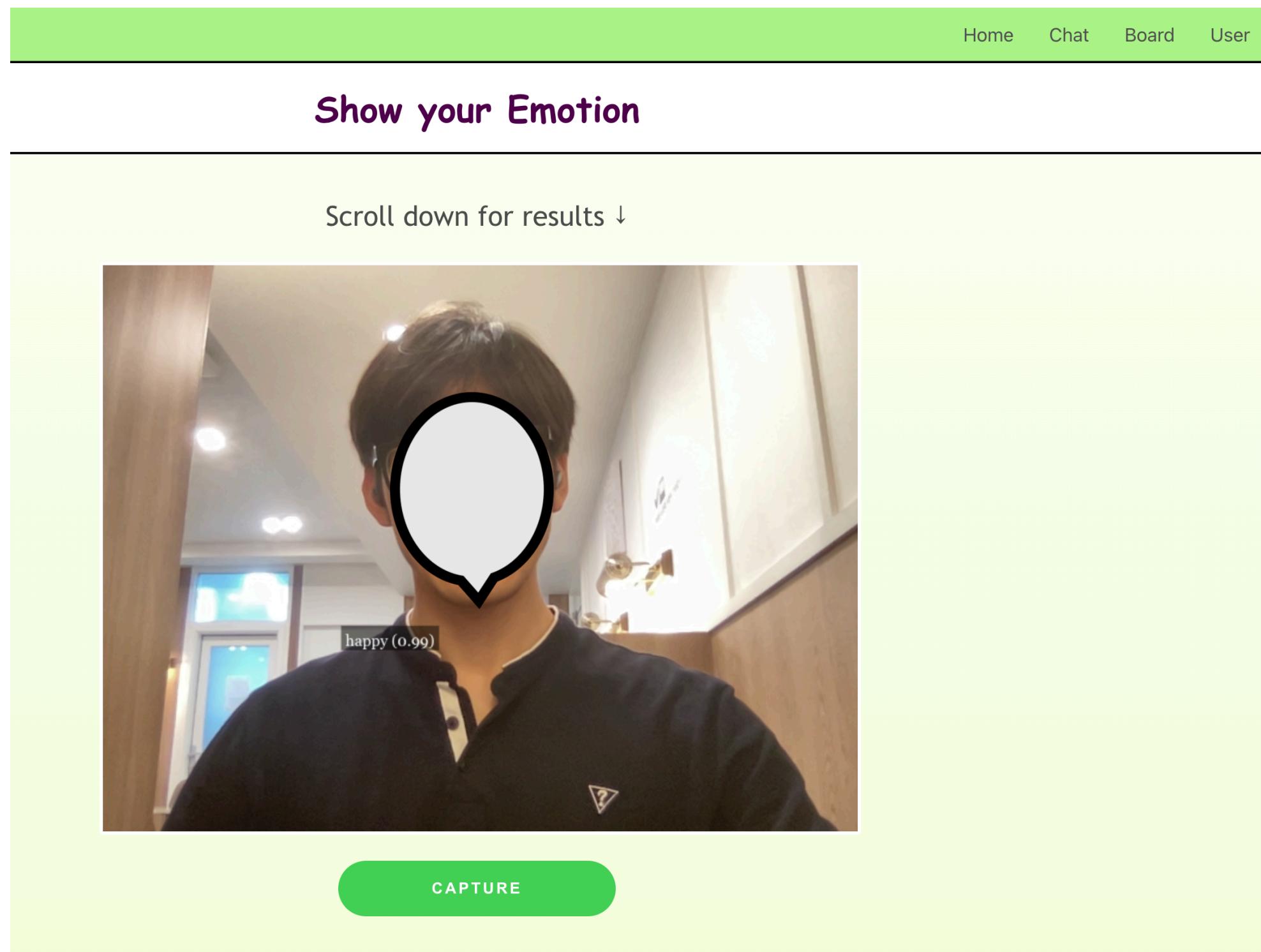
Login: 로그인이 성공적으로 수행되었다면 서버에서는 jwt 토큰을 발급하고 클라이언트 웹 브라우저의 local storage에 저장합니다.

# 구현 모습 - 안면 인식

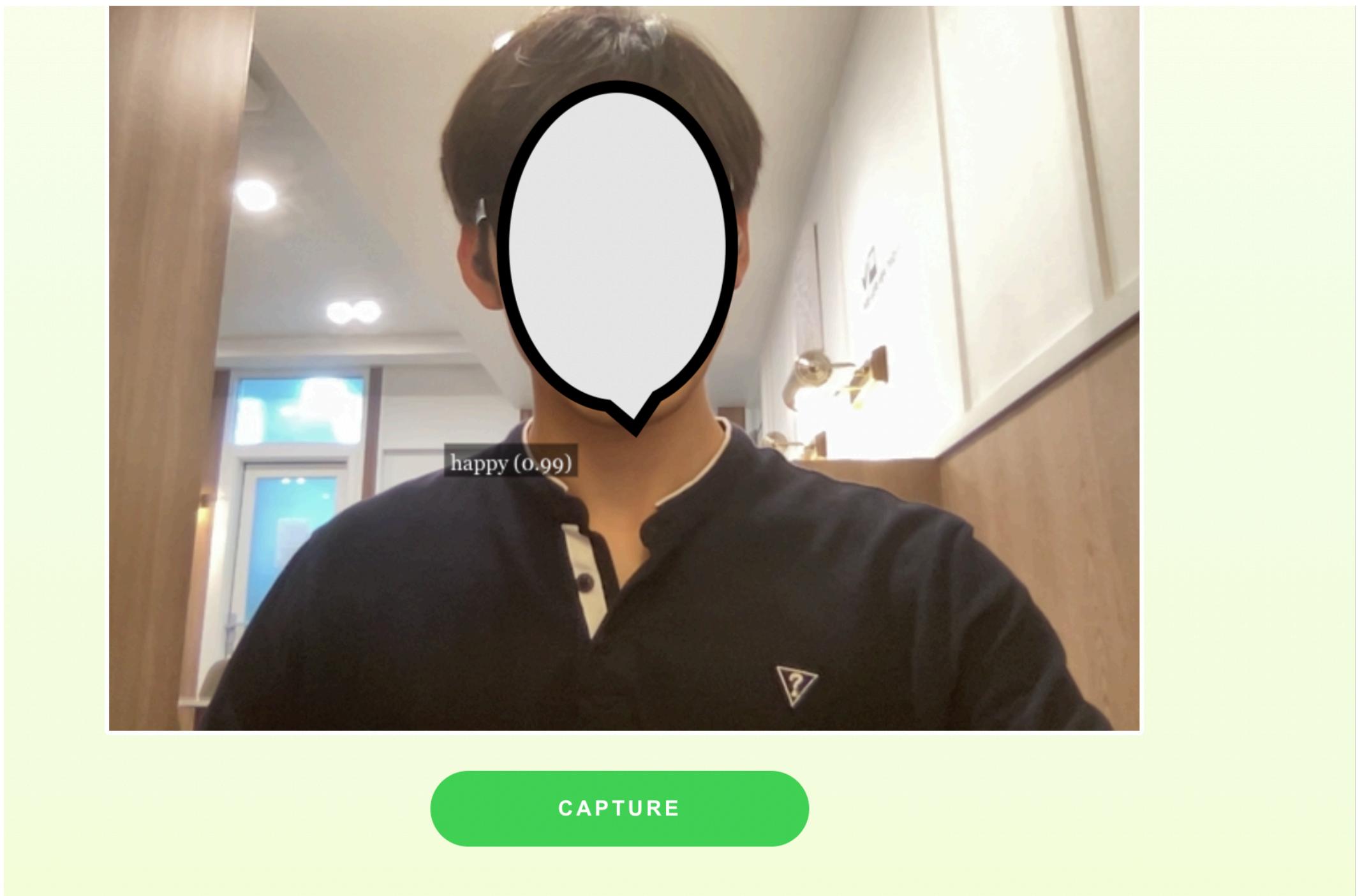
Home Chat Board User

Show your Emotion

Scroll down for results ↓



CAPTURE



I think You look happy 😊

# 구현 설명 - 안면 인식

```
import * as faceapi from "face-api.js";

export const loadModels = () => {
  const MODEL_URL = `${process.env.PUBLIC_URL}/models`;
  //console.log("Model_URL: ", MODEL_URL);

  return Promise.all([
    faceapi.nets.tinyFaceDetector.loadFromUri(MODEL_URL),
    faceapi.nets.faceLandmark68Net.loadFromUri(MODEL_URL),
    faceapi.nets.faceRecognitionNet.loadFromUri(MODEL_URL),
    faceapi.nets.faceExpressionNet.loadFromUri(MODEL_URL),
    faceapi.nets.ageGenderNet.loadFromUri(MODEL_URL),
  ]);
};

export const detectFaces = async (image) => {
  if (!image) {
    return;
  }

  try {
    const imgSize = image.getBoundingClientRect();
    const displaySize = { width: imgSize.width, height: imgSize.height };
    if (displaySize.height === 0) {
      return;
    }
    //console.log("detectFace is called");
    const faces = await faceapi
      .detectAllFaces(
        image,
        new faceapi.TinyFaceDetectorOptions({ inputSize: 320 })
      )
      .withFaceLandmarks()
      .withFaceExpressions()
      .withAgeAndGender();
    return faceapi.resizeResults(faces, displaySize);
  } catch (e) {
    console.error("faceApi error occurred: ", e);
  }
};
```

```
export const drawResults = async (image, canvas, results, type) => {
  //console.log("drawResult is called");
  try {
    if (image && canvas && results) {
      const imgSize = image.getBoundingClientRect();
      const displaySize = { width: imgSize.width, height: imgSize.height };
      faceapi.matchDimensions(canvas, displaySize);
      canvas.getContext("2d").clearRect(0, 0, canvas.width, canvas.height);
      const resizedDetections = faceapi.resizeResults(results, displaySize);

      switch (type) {
        case "landmarks":
          faceapi.draw.drawFaceLandmarks(canvas, resizedDetections);
          break;
        case "expressions":
          faceapi.draw.drawFaceExpressions(canvas, resizedDetections);
          break;
        case "box":
          faceapi.draw.drawDetections(canvas, resizedDetections);
          break;
        case "boxLandmarks":
          faceapi.draw.drawDetections(canvas, resizedDetections);
          faceapi.draw.drawFaceExpressions(canvas, resizedDetections);
          faceapi.draw.drawFaceLandmarks(canvas, resizedDetections);
          break;
        default:
          break;
      }
    }
  } catch (e) {
    console.error("drawResults error occurred: ", e);
  }
};
```

안면 인식 과정은 크게 2가지 요소로 구성되어 있습니다.

카메라 모듈, 인식 모듈입니다.

위의 코드는 인식 모듈로써 노트북에 내장되어 있는 웹캠으로부터 전달받은 실시간 안면 데이터를 분석하여 감정을 추출하는 코드입니다.

# 구현 설명 - 안면 인식

안면 인식 과정은 크게 2가지 요소로 구성되어 있습니다.

카메라 모듈, 인식 모듈입니다.

오른쪽의 코드는 노트북에 내장되어 있는 웹캠 기능을 불러와 카메라로

안면을 실시간으로 촬영하는 코드입니다.

```
const Camera = () => {
  const camera = useRef();
  const cameraCanvas = useRef();

  const [photo, setPhoto] = useRecoilState(faceImage);
  const [results, setResults] = useRecoilState(detectResult);
  const [captured, setCaptured] = useRecoilState(isCaptured);
  const [needStop, setNeedStop] = useRecoilState(needDetectStop);
  const navigate = useNavigate();

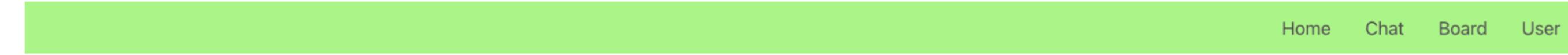
  async function getFaces() {
    //console.log("getFaces is called");

    if (camera.current !== null) {
      try {
        const faces = await detectFaces(camera.current.video);
        await drawResults(
          camera.current.video,
          cameraCanvas.current,
          faces,
          "expressions" // "boxLandmarks"
        );
        setResults(faces);
      } catch (e) {
        console.error(e);
      }
    }
  }

  const clearOverlay = (canvas) => {
    console.log("clearOverlay called");
    canvas.current
      .getContext("2d")
      .clearRect(0, 0, canvas.width, canvas.height);
  };

  useEffect(() => {
    if (needStop) {
      return;
    }
    if (!photo && camera !== null) {
      try {
        const drawOverlay = () => {
          getFaces();
        };
        const loopCall = setInterval(drawOverlay, 200);
        return () => {
          //clearOverlay(cameraCanvas);
          clearInterval(loopCall);
        };
      } catch (e) {
        console.error("camera useEffect error occurred: ", e);
      }
    } else {
      //return clearOverlay(cameraCanvas);
    }
  });
}
```

# 구현 모습 - 챗봇



i think i'm not happy. do you know about this feeling?

Yes, I understand the feeling of not being happy. It is common for people to experience times when they feel unhappy or dissatisfied with their lives. It can be caused by various factors such as stress, relationships, work, health, or personal circumstances. If you want to talk more about your feelings, I'm here to listen.

can you suggest some music that i can change my mood?

Of course! Here are a few different genres and artists that may help to uplift or change your mood: 1. Pop: - "Happy" by Pharrell Williams - "Dancing Queen" by ABBA - "Don't Stop Me Now" by Queen 2. Rock: - "Mr. Brightside" by The Killers - "Eye of the Tiger" by Survivor - "Livin' on a Prayer" by Bon Jovi 3. R&B/Soul: - "I Wanna Dance with Somebody" by Whitney Houston - "Uptown Funk" by Mark Ronson ft. Bruno Mars - "Ain't No Mountain High Enough" by Marvin Gaye and Tammi Terrell 4. Electronic/Dance: - "Levels" by Avicii - "One More Time" by Daft Punk - "Wake Me Up" by Avicii 5. Indie/Alternative: - "Take Me Out" by Franz Ferdinand - "Kids" by MGMT - "Pumped Up Kicks" by Foster the People Remember, music is subjective, so explore these suggestions and find what resonates with you the most.

ask me anything...

# 구현 설명 - 챗봇

```
export default function Prompts(emotion) {
  const basePromts = [`I'm feeling ${emotion} now.`];

  const additonalPromts = [
    "Any recommendations on food to refresh my feeling?",
    "Can you suggest a travel destination that can improve my mood?",
    "What exercise will help me?",
    "Please give me music that suits my mood",
    "Please recommend a movie that makes me feel better",
    "What is the cause of my current mood?",
    "Please introduce me to a helpful phrase",
    "Are you feeling the same way?",
    "Do you understand what this feeling like?",
    "What do I need?",
  ];

  let random_index = Math.floor(Math.random() * additonalPromts.length);
  let rand_promt = additonalPromts[random_index];
  let resPromt = basePromts + " " + rand_promt;

  //console.log(resPromt);
  return resPromt;
}

Prompts();
```

사용자의 안면 인식을 통해 ‘happy’, ‘sad’, ‘surprised’ 등의 키워드를 추출해내고 이 키워드를 랜덤으로 조합하여 하나의 프롬프트를 생성해냅니다.  
이 프롬프트는 후에 gpt에게 전달되어 감정과 관련된 조언을 받아옵니다.

# 구현 설명 - 챗봇

```
@api_view(['POST'])
def ChatHandler(request):
    # front로부터 전달받은 chat 객체 parsing
    chat = request.data

    try:
        # print("requested chat: ", chat)
        user_id = chat['user_id']
        content = chat['content']

        # DB에 저장할 객체 생성
        chatObj = Chats(sender_id=user_id, receiver_id="bot",
                         content=content) # time_stamp=timezone.now()

        # DB에 저장
        chatObj.save()

        # gpt request
        openai.api_key = getEnvAttr('OPENAI_API_KEY')
        messages = [
            {"role": "user", "content": content}
        ]

        completion = openai.ChatCompletion.create([
            model="gpt-3.5-turbo",
            messages=messages
        ])

        chat_response = completion.choices[0].message.content # "hello from jango"
        # print(f'ChatGPT: {chat_response}')

        # response 객체 생성
        resObj = Chats(sender_id="bot", receiver_id=user_id,
                        content=chat_response) # time_stamp=timezone.now()

        # DB 저장
        resObj.save()

        # response 객체 생성 후 전달. text 수정 필요
        resForFront = {'text': chat_response, 'isUser': False}
        return Response(status=status.HTTP_200_OK, data=resForFront)
    except Exception as e:
        return JsonResponse({'error': str(e)}, status=500)
```

```
class ChatView(APIView):

    # sender가 userA이면서 receiver가 Bot인 경우와 sender가 Bot이면서 receiver가 userA인 경우.. Query
    # (sender == userA && receiver == Bot || sender == Bot && receiver == userA)

    def post(self, request, *args, **kwargs):
        try:
            # print("req data: ", request.data)
            user_id = request.data['user_id'] # 클라이언트 사용자의 ID
            # print("user_id: ", user_id)

            user_to_bot_chats = Chats.objects.filter(
                sender_id=user_id, receiver_id='bot')
            # print("user_to_bot_chats: ", user_to_bot_chats)

            # sender가 Bot이면서 receiver가 userA인 경우 데이터 가져오기
            bot_to_user_chats = Chats.objects.filter(
                sender_id='bot', receiver_id=user_id)

            # print("sender_bot: ", bot_to_user_chats)

            # 두 QuerySet을 하나로 합치기
            combined_chats = user_to_bot_chats | bot_to_user_chats

            # 시간순으로 정렬
            # sorted_chats = combined_chats.order_by('time_stamp')
            # print("sorted_chats: ", combined_chats)

            serializer = ChatsListSerializer(
                combined_chats,
                many=True,
                context={'user_id': user_id})
        )
            return Response(serializer.data)
        except Exception as e:
            return JsonResponse({'error': str(e)}, status=500)
```

Open AI의 chatgpt 3.5 turbo model을 활용하여 사용자의 감정을 추출하여 랜덤으로 생성한 프롬프트로 답변을 받아오는 코드입니다.

# 회고

- 아직 버그도 많고 UI도 부족한거 같습니다.
- 조금 더 기능을 고도화해서 서비스 호스팅 및 운영까지 해보고 싶은 생각이 들었습니다.
- 현재는 프론트를 react로 구현해놓았는데 React Native로 migration해서 모바일 어플리케이션으로 Android, ios에 배포해보자는 목표를 가지게 되었습니다.

# 참조 자료

- <https://github.com/justadudewhohacks/face-api.js>
- <https://platform.openai.com/docs/api-reference>