

10주차(11.04~11.10) 주간 보고서

1. 프론트엔드

스택 바 그래프

코드 정리

서버에서 받는 데이터 수정

이름표 제작

버전 1

문제점

버전 2

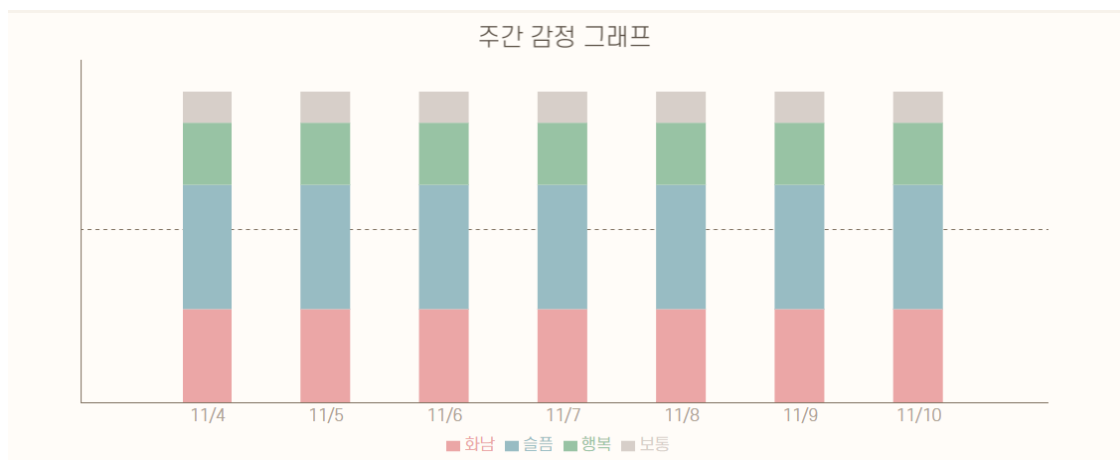
감정에 따른 모달 디자인 변경

방법

2. 백엔드

1. 프론트엔드

스택 바 그래프



```
import { BarChart, Bar, XAxis, Tooltip, Legend } from 'recharts'

const data = [
  {
    day: "11/4",
```

```

    보통: Epercent[0],
    행복: Epercent[1],
    슬픔: Epercent[2],
    화남: Epercent[3]
  },
  {
    day: "11/5",
    보통: Epercent[0],
    행복: Epercent[1],
    슬픔: Epercent[2],
    화남: Epercent[3]
  },
  {
    day: "11/6",
    보통: Epercent[0],
    행복: Epercent[1],
    슬픔: Epercent[2],
    화남: Epercent[3]
  },
  {
    day: "11/7",
    보통: Epercent[0],
    행복: Epercent[1],
    슬픔: Epercent[2],
    화남: Epercent[3]
  },
  {
    day: "11/8",
    보통: Epercent[0],
    행복: Epercent[1],
    슬픔: Epercent[2],
    화남: Epercent[3]
  },
  {
    day: "11/9",
    보통: Epercent[0],
    행복: Epercent[1],
    슬픔: Epercent[2],

```

```

        화남: Epercent[3]
      },
      {
        day: "11/10",
        보통: Epercent[0],
        행복: Epercent[1],
        슬픔: Epercent[2],
        화남: Epercent[3]
      }
    ];

    const formatTooltip = (value) => {
      return `${value.toLocaleString()}%`;
    };

    <div>
      <BarChart
        className={styles.Chart}
        width={800}
        height={410}
        data={data}
        margin={{
          top: 40,
          right: 0,
          left: -35,
          bottom: 5,
        }}
      >
        <XAxis dataKey="day" tickLine={false} stroke='' />
        <Tooltip formatter={formatTooltip} />
        <Legend />
        <Bar dataKey={"화남"} stackId="a" fill="#EBA6A6" />
        <Bar dataKey={"슬픔"} stackId="a" fill="#98BCC3" />
        <Bar dataKey={"행복"} stackId="a" fill="#98C3A4" />
        <Bar dataKey={"보통"} stackId="a" fill="#D7CFC9" />
      </BarChart>
    </div>

```

코드 정리

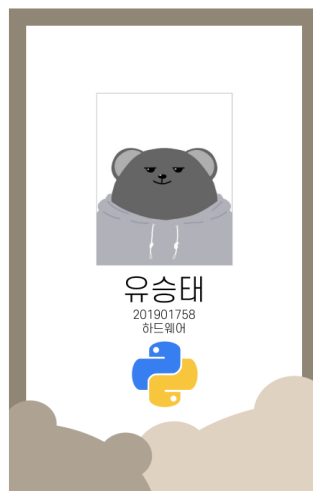
필요 없어진 코드와 변수들을 정리

서버에서 받는 데이터 수정

```
useEffect(() => {
  axios.get(URL, {
    headers: {
      'x-access-token': userToken
    }
  })
  .then(response => {
    console.log('일기 정보', response.data);
    setEpercent(response.data.map(entry => entry.emotio
    setstatuses(response.data.map(entry => entry.emotio
    setcontents(response.data.map(entry => entry.conten
    setDate(response.data.map(entry => entry.writeAt));
  });
}, [userToken]);
```

이름표 제작

버전 1



문제점

1. 너무 작음
2. 이름이 너무 작아서 이름표 역할을 못함

버전 2



감정에 따른 모달 디자인 변경

방법

일단 태수님이 감정 분석을 GPT로 돌려서 보내주시면

감정에 맞게 이미지와 애니메이션을 넣기

애니메이션은

슬픔: 물방울 여러 개 내리기 💧

화남: 불 모양 이모지가 올라가기 🔥

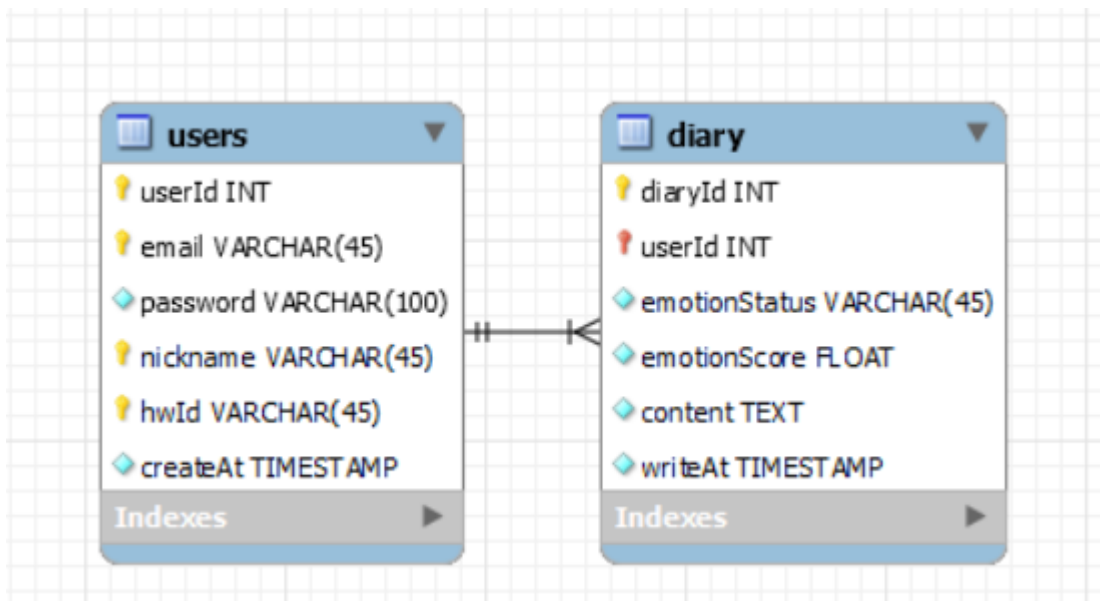
기쁨: 꽃 같은거 날라다니기 🌸

보통: 비누방울이나 풀 날라다니기 🫧, 🌿

2. 백엔드

- 데이터 베이스 수정

기존의 consolation, summarize 테이블을 삭제



- indexdao 및 indexcontroller 코드 수정


```
// 일기 삽입
exports.insertDiary = async function (
  userId,
  emotionStatus,
  emotionScore,
  content
) {
  let connection;
  try {
    connection = await pool.getConnection(async (conn) => conn);
    const insertDiaryQuery =
      "INSERT INTO diary (userId, emotionStatus, emotionScore, content) VALUES (?, ?, ?, ?)";
    const insertDiaryParams = [userId, emotionStatus, emotionScore, content];

    const [row] = await connection.query(insertDiaryQuery, insertDiaryParams);
    return row;
  } catch (err) {
    console.error(` ### insertDiary Query/DB error ### \n ${err}`);
    return false;
  } finally {
    if (connection) connection.release();
  }
};
```

```
// 일기 생성
exports.createDiary = async function (req, res) {
  try {
    // 요청 본문에서 필요한 값을 비구조화 할당으로 추출
    const { userId, emotionStatus, emotionScore, content } = req.body;

    // 입력값 검증
    if (!userId || !emotionStatus || !emotionScore || !content === undefined) {
      return res.status(400).send({
        isSuccess: false,
        code: 400,
        message: "입력값이 누락되었습니다.",
      });
    }

    // DB에 일기 저장
    const insertDiaryRow = await indexDao.insertDiary(
      userId,
      emotionStatus,
      emotionScore,
      content
    );
  }
};
```

몇 주간 진행이 안된 프론트엔드와 백엔드 연동 성공

- 가비아에서 내가 사용할 인스턴스에 dns 서비스를 이용.
- win-scp 를 이용해 가상의 컴퓨터와 내 인스턴스를 리버스 프록시를 이용해 연동.
- pm2 start back/src/index.js 명령어를 사용해 무중단 배포 실행.