

CSS를 적용후 변경된 코드입니다

src/Meta.js

```
import React from "react";
import { Helmet, HelmetProvider } from "react-helmet-async";

const Meta = (props) => {
  return (
    <HelmetProvider>
      <Helmet>
        <meta charset="utf-8" />
        <title>{props.title}</title>
        { /* SEO 태그 */ }
        <meta name="description" content={props.description} />
        <meta name="keywords" content={props.keywords} />
        <meta name="author" content={props.author} />
        <meta property="og:type" content="website" />
        <meta property="og:title" content={props.title} />
        <meta property="og:description" content={props.description} />
        <meta property="og:image" content={props.image} />
        <meta property="og:url" content={props.url} />
        <link rel="preconnect" href="https://fonts.googleapis.com" />
        <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
        <link
          href="https://fonts.googleapis.com/css2?family=Gugi&family=Noto+Sans+KR:wght@100;300;400;500&display=swap"
          rel="stylesheet"
          type="text/css"
        />

        { /* Helmet 안에서 CSS적용하기 */ }
        <style type="text/css">{`
          *{
            list-style: none;
          }
          body {
            margin: 0;
            padding: 0;
          }

          `}</style>

        { /* 추가적으로 적용해야 할 외부 js나 css로 여기서 명시할 수 있다. */ }
        { /*font awesome 참조*/ }
        <script
          src="https://kit.fontawesome.com/aa134343d6.js"
          crossorigin="anonymous"
        ></script>
      </Helmet>
    </HelmetProvider>
  );
};
```

```
};
/* 이게 props */
Meta.defaultProps = {
  title: "useage of OpenAPI",
  description: "OpenAPI 활용 제출물입니다.",
  keywords: "OpenAPI",
  author: "최수진,박세영",
  url: window.location.href,
};

export default Meta;
```

src/pages/misea.js

```
/**
 * @filename: misae.js
 * @description: 화면에 실질적으로 보여지는 컴포넌트 (액션함수를 dispatch한다)
 * @author: 최수진(sujin971008@gmail.com), 박세영(qkrtpdud9899@gmail.com)
 */
import React from "react";
import styled from "styled-components";
import { useSelector, useDispatch } from "react-redux";

/* slice */
import { getInfo } from "../slices/misaeSlice";

import Spinner from "../components/Spinner";
import Error from "../components/Error";

/* styledComponent */
const Div = styled.div`
  width: 100%;
  display: flex;
  flex-direction: column;
  align-items: center;
  .area {
    width: 50%;
    display: flex;
    justify-content: space-around;
    border-radius: 3%;
    padding-bottom: 3%;
    h3 {
      padding-top: 20px;
    }
    .dustArea {
      width: 50%;
      text-align: center;
      .integrative {
        p{
```

```

        font-size:30px;
        font-weight:bold;
    }
}
.dust {
    li {
        font-size: 20px;
        font-weight: bolder;
    }
}
.etc {
    width: 100%;
}
}
}
}
.textArea {
    margin-top: 4%;
    text-align: center;
    opacity: 0.6;
    background-color: lightgray;
    color: black;
    p {
        font-size: 12px;
    }
    div {
        font-size: 13px;
        i {
            color: red;
        }
        span {
            font-weight: bold;
        }
    }
}
};
const SelectContainer = styled.div`
    position: sticky;
    top: 0;
    margin: 0;
    display: flex;
    align-items: flex-start;
    padding-top: 10%;
    select {
        border-radius: 8px;
        margin-right: 15px;
        font-size: 16px;
        padding: 5px 5px;
        border: 3px solid black;
        option {
            background-color: black;
            color: #fff;
        }
    }
}
`;

```

```

const Misae = () => {
  React.useEffect(() => console.clear(), []);
  // hook을 통해 slice가 관리하는 상태값 가져오기
  const { items, loading, error } = useSelector((state) => state.getInfo);

  /* Grade 함수1 */
  function Grade(x) {
    let state = null;
    if (x === "1") {
      state = " 좋음";
    } else if (x === "2") {
      state = " 보통";
    } else if (x === "3") {
      state = " 나쁨";
    } else if (x === "4") {
      state = " 매우나쁨";
    } else {
      state = " 오류";
    }
    return state;
  }

  /* Grade값 함수2 */
  function Emoji(y) {
    let state = null;
    if (y === "1") {
      state = "😊";
    } else if (y === "2") {
      state = "😐";
    } else if (y === "3") {
      state = "😬";
    } else if (y === "4") {
      state = "😱";
    } else {
      state = "오류";
    }
    return state;
  }

  // dispatch 함수 생성
  const dispatch = useDispatch();

  /* 상태값을 '중구'로 기본값을 잡아줌으로써, 페이지가 열리자마자 정보를 보여주게 된다
  (앱과 같은 느낌을 주고 싶었다) */
  const [stationName, setStationName] = React.useState("중구");

  // 컴포넌트가 마운트되면 데이터 조회를 위한 액션함수를 디스패치 함
  React.useEffect(() => {
    dispatch(getInfo());
  }, [dispatch, stationName]);

  /* 자치구 선택 시 Event */
  const onSelectChange = React.useCallback((e) => {

```

```

    e.preventDefault();
    // 드롭다운의 입력값 취득
    const current = e.target;
    const value = current[current.selectedIndex].value;
    setStationName((stationName) => value);
  }, []);

  //배경색을 바꾸기위해 참조
  const bgRef = React.useRef();

  return (
    <Div>
      <Spinner visible={loading} />
      <h1>서울특별시 대기질 현황</h1>
      {error ? (
        <Error error={error} />
      ) : (
        <div className="area" ref={bgRef}>
          {/* 검색 조건 드롭다운 박스 */}
          <SelectContainer>
            <select name="location" onChange={onSelectChange}>
              {/* items의 지역명을 반복 돌려서 option 생성 */}
              <option value="">-- 자치구 선택 --</option>
              {items &&
                items.map((v, i) => {
                  return (
                    <option value={v.stationName} key={i}>
                      {v.stationName}
                    </option>
                  );
                })}
            </select>
          </SelectContainer>
          {items &&
            items.map((v, i) => {
              //v.khaiGrade 값에 따른 배경색 분기
              if (v.khaiGrade === "1") {
                bgRef.current.style.backgroundColor = "#01DF01";
              } else if (v.khaiGrade === "2") {
                bgRef.current.style.backgroundColor = "#F7D358";
              } else if (v.khaiGrade === "3") {
                bgRef.current.style.backgroundColor = "#FD9903";
              } else if (v.khaiGrade === "4") {
                bgRef.current.style.backgroundColor = "#FF1C11";
              } else {
                bgRef.current.style.backgroundColor = "#fff";
              }
            })
            return (
              v.stationName === stationName && (
                <div className="dustArea">
                  <h3>{v.stationName}</h3>
                  <div className="integrative">
                    {/* 통합지수 */}
                    <p>

```

```

        {v.stationName === stationName
          ? Grade(v.khaiGrade)
          : ""}
      </p>
      { /* 통합지수 이모지 */ }
      <span style={{ fontSize: "120px" }}>
        {v.stationName === stationName
          ? Emoji(v.khaiGrade)
          : ""}
      </span>
      { /* 미세먼지 */ }
      <div className="dust">
        {v.stationName === stationName ? (
          <ul>
            <li>
              미세먼지: {v.pm10Value}µg/m³
              {Emoji(v.pm10Grade)}
            </li>
            <li>
              초미세먼지: {v.pm25Value}µg/m³
              {Emoji(v.pm25Grade)}
            </li>
          </ul>
        ) : (
          <ul></ul>
        )}
      </div>
    </div>
    { /* 기타 대기오염지수 테이블 */ }
    <table className="etc">
      {v.stationName === stationName ? (
        <thead>
          <tr>
            <th>아황산가스</th>
            <td>
              {Grade(v.so2Grade)}
              {Emoji(v.so2Grade)}
            </td>
            <td>{v.so2Value}ppm</td>
          </tr>
          <tr>
            <th>일산화탄소</th>
            <td>
              {Grade(v.coGrade)}
              {Emoji(v.coGrade)}
            </td>
            <td>{v.coValue}ppm</td>
          </tr>
          <tr>
            <th>오존</th>
            <td>
              {Grade(v.o3Grade)}
              {Emoji(v.o3Grade)}
            </td>

```

```

        <td>{v.o3Value}ppm</td>
      </tr>
      <tr>
        <th>이산화질소</th>
        <td>
          {Grade(v.no2Grade)}
          {Emoji(v.no2Grade)}
        </td>
        <td>{v.no2Value}ppm</td>
      </tr>
    </thead>
  ) : (
    <thead></thead>
  )}
</table>
</div>
)
);
}}}
</div>
)}
<div className="textArea">
  <div>
    <i
      className="fa-solid fa-triangle-exclamation"
      aria-hidden="true"
    ></i>
    <span>주의사항</span>
    <p>
      해당 기관이 제공하는 자료는 “인증을 받지 않은 실시간자료”이므로 자료
      오류 및 표출방식에 따라 값이 다를 수 있습니다.
    </p>
  </div>
  <p>자료 출처: 공공데이터포털을 통한 환경부/한국환경공단 에어코리아 </p>
</div>
</Div>
);
};

export default React.memo(Misae);

```