

Fumease



소스코드 : <https://github.com/Kbomi16/fumease>

주제

- AI로 향수 쇼핑할 수 있는 웹사이트 (팀 프로젝트-3명) perfume + Ease
향수 + 쉽게

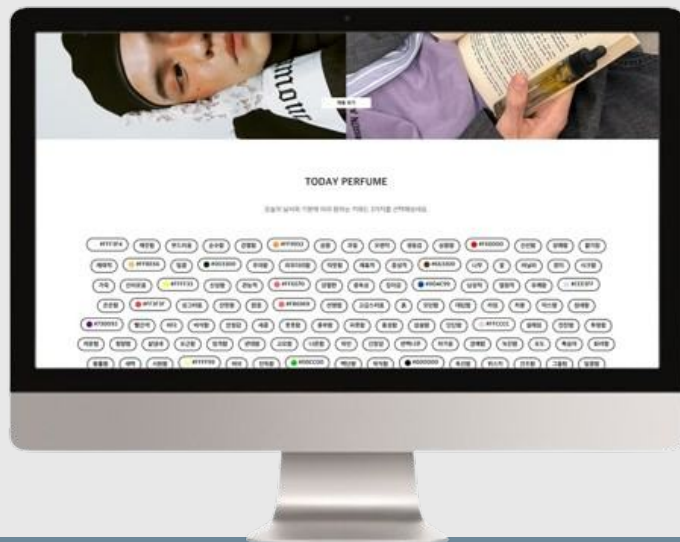
개발 기간 : 2023.10 ~ 2023.12

주요기능

1. 키워드와 색을 선택해서 본인 취향에 맞는 향수를 추천해주는 AI
2. 추천된 향수가 나오면 구매할 수 있는 향수 쇼핑 기능(장바구니 기능)
3. 향수 정보를 제공
4. 로그인/회원가입

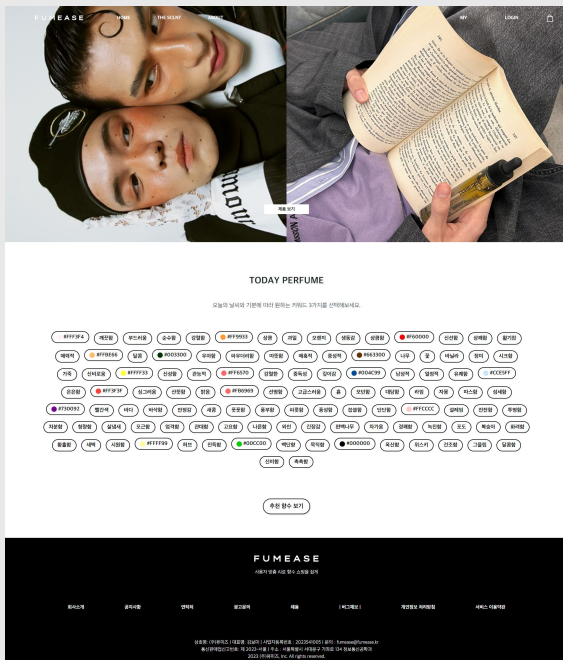
Stack & Tools

- React, React Bootstrap
- MySQL
- NodeJS
- Express
- OpenAI



주요 기능

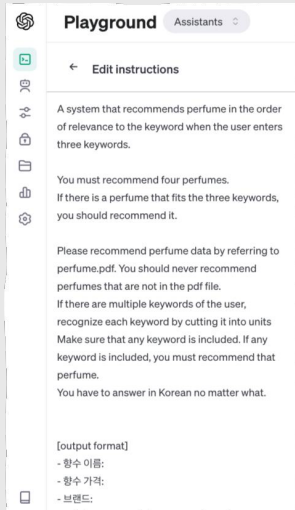
(1) 키워드와 색을 선택해서 본인 취향에 맞는 향수를 추천해주는 AI



Open AI 의 assistant 를 통해 향수 DB를 저장해놓고, 키워드를 선택하면 해당 키워드에 맞는 향수를 추천해주는 AI

키워드가 한 묶음으로 나오는 오류를 발견했다.

해결하기 위해 데이터가 배열인지 확인하고, split으로 모든 키워드를 쉼표를 기분으로 단어를 분할한다. Set 함수를 통해 중복되는 키워드를 제거하여 깔끔하고 직관적으로 키워드를 볼 수 있다.



OpenAI의 assistant를 통해 명령어를 입력시킨다.

output format을 지정하여 내가 원하는 답변의 형태로 나오게 설정했다.

DB 정보를 학습시켜 사용자가 입력한 키워드에 맞는 향수를 찾도록 한다.

처음에 output format을 영어로 했더니 추천된 향수가 보이지 않는 오류를 발견했다.

해결하기 위해 터미널을 보았더니 번역해서 나와서 Perfume name이 향수 이름이 아닌 향수명으로 나오는 것을 확인했다. output format만 한국어로 설정하여 gpt 답변에서 정확히 추출될 수 있도록 했다.

```
instructions: 'A system that recommends perfume in the order of relevance to the keyword when the user enters three keywords.\n' +
'\n' +
'You must recommend four perfumes.\n' +
'If there is a perfume that fits the three keywords, you should recommend it.\n' +
'\n' +
'Please recommend perfume data by referring to perfume.pdf. You should never recommend perfumes that are not in the pdf file.\n' +
'If there are multiple keywords of the user, recognize each keyword by cutting it into units.\n' +
'Make sure that any keyword is included. If any keyword is included, you must recommend that perfume.\n' +
'You have to answer in Korean no matter what.\n' +
'\n' +
'\n' +
'[output format]\n' +
'- 향수 이름:\n' +
'- 향수 가격:\n' +
'- 브랜드:',
tools: [ { type: 'retrieval' } ],
file_ids: [ 'file-sp2XazDH2ydX03G3ad046JLI' ],
metadata: {}
```

```
First message content: {
  type: 'text',
  text: {
    value: '- 향수 이름: 잉글리쉬 페어 앤 프리지아 코롱\n' +
'- 향수 가격: 220,000원\n' +
'- 브랜드: 조말론[ 14↑source] \n' +
'\n' +
'- 향수 이름: 우드 세이지 앤 씨 솔트 코롱\n' +
'\n' +
'- 향수 가격: 220,000원\n' +
'- 브랜드: 조말론[ 15↑source] \n' +
'\n' +
'- 향수 이름: SIMPLE GARDEN\n' +
'- 향수 가격: 158,000원\n' +
'- 브랜드: 논픽션[ 16↑source] \n' +
'\n' +
'- 향수 이름: 비자리아 오드퍼퓸\n' +
'- 향수 가격: 234,000원\n' +
'- 브랜드: 산타마리아노벨라[ 17↑source] ',
```

사용자가 선택한 키워드를 백엔드에 전달 후, OpenAI에서 설정한 instructions가 실행되며 향수를 추천한다.

시간이 지난 후, 터미널에 gpt가 내놓은 답변이 뜬다.

```
// GPT의 텍스트를 줄 단위로 분리하고, 각 줄을 배열로 변환함.
function textToArray(text) {
  const lines = text.split("\n");
  const perfumes = [];
  let currentPerfume = {};

  lines.forEach((line, index) => {
    if (line.includes("- 향수 이름:")) {
      if (Object.keys(currentPerfume).length !== 0) {
        perfumes.push(currentPerfume);
        currentPerfume = {};
      }
      currentPerfume["name"] = line.split(": ")[1].trim();
    } else if (line.includes("- 향수 가격:")) {
      currentPerfume["price"] = line.split(": ")[1].trim();
    } else if (line.includes("- 브랜드:")) {
      currentPerfume["brand"] = line.split(": ")[1].trim();
    }
  });

  // 마지막 향수 정보를 배열에 추가
  if (Object.keys(currentPerfume).length !== 0) {
    perfumes.push(currentPerfume);
  }

  return perfumes;
}
```

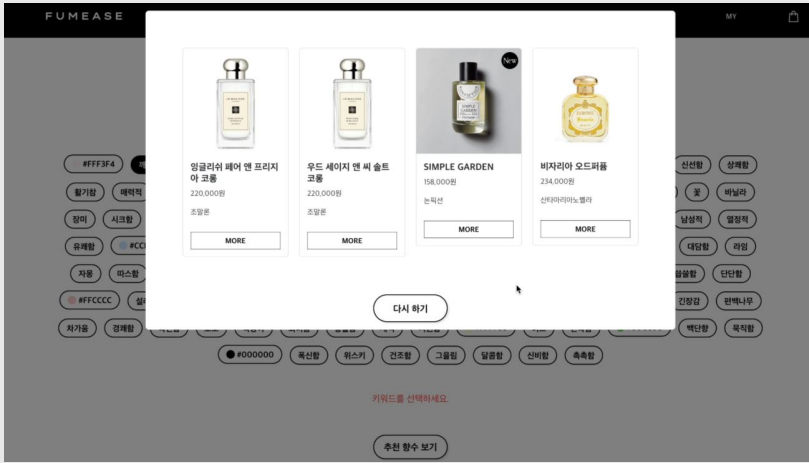
```
do {
  list = await requestGPT();
  const perfumeNames = list.map(perfume => perfume.name);
  console.log("GPT가 추천한 향수 이름: " + list.length);
  perfumes = await Promise.all(
    perfumeNames.map(perfumeName => {
      if (perfumeName) {
        return Perfume.findOne({
          attributes: {
            "f_id",
            "f_name",
            "f_price",
            "f_img",
            "f_brand",
            "f_keyword",
          },
          where: {
            f_name: perfumeName,
          },
        });
      }
    })
  );
} while (perfumes.length > 0);
console.log("추출된 향수 리스트: ", perfumes.length);
//향수 한 개라도 추출되면 계속 반복..
```

gpt가 내놓은 답변을 줄 단위로 분리하고, 반복문 안에서 각 줄이 “향수 이름” (perfume name), “향수 가격” (perfume price), “브랜드” (brand)와 같은 특정 키워드를 포함하는지 확인한다.

포함하고 있다면, 해당 정보를 추출하여 currentPerfume 객체에 할당
새로운 향수 이름이 있으면 기본 currentPerfume 객체를 perfumes 배열에
추가하고 currentPerfume 을 빈 객체로 초기화하여 새로운 향수 정보를 저장한다.

프론트 화면에서 추천 향수가 뜨지 않는 오류를 발견했다. 확인해보니 gpt의 오류로 인해 터미널에서 제대로 된 답변이 나오지 않을 때가 있다는 것이었다.

해결하기 위해 Do-while 문을 통해 향수의 배열이 무조건 하나라도 나올 때까지 반복하여 절대 빈 화면이 나오지 않도록 했다.



GPT 답변에서 모달을 통해 향수 이름, 가격, 브랜드를 추출하여 추천 향수를 띄운다.

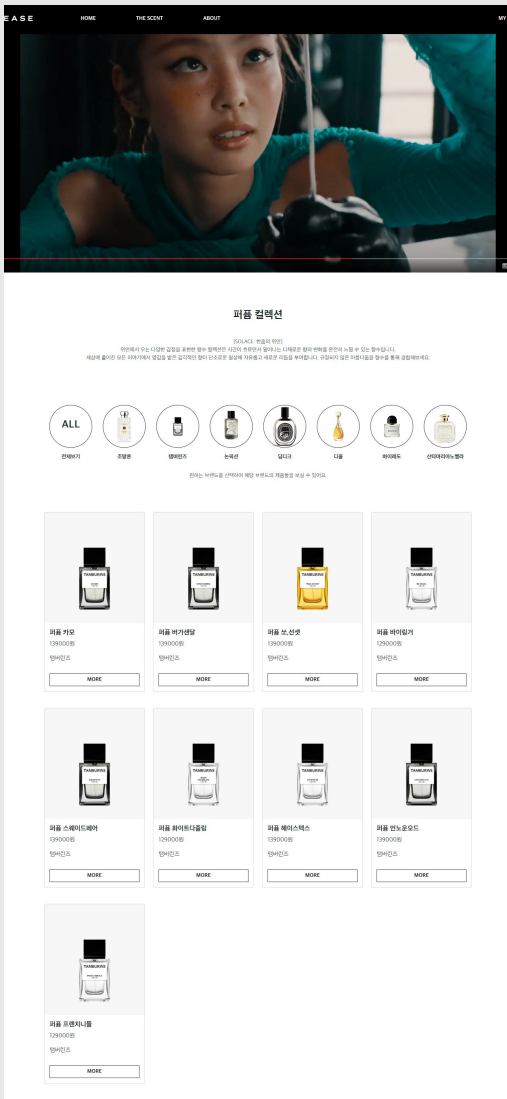
MORE 를 누르면 해당 향수의 id를 찾아 제품 상세 페이지로 이동한다.



추천 버튼을 누르면 지루하지 않게 로딩 스피너와 2초마다 변하는 메시지를 통해 로딩 화면을 구현했다.

주요 기능

(2) 추천된 향수가 나오면 구매할 수 있는 향수 쇼핑 기능



React-youtube npm 설치하여 유튜브 영상을 상위에 뜨게 설정하였다.

향수 브랜드의 종류를 보여준다.

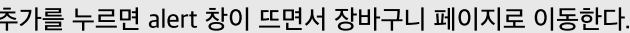
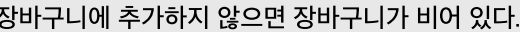
스크롤 페이지를 이용하여 모든 상품을 다 띄울 수 있도록 했다. 클릭 이벤트를 이용하여 브랜드 별 제품을 모아볼 수 있도록 필터링 설정했다.

페이지네이션 기능을 포함하고 클릭이벤트를 지정하여 원하는 브랜드를 선택할 때 모든 페이지가 로딩되지 않으면 원하는 정보가 나오지 않는 오류를 발견했다.

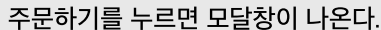
해결하기 위해 페이지네이션을 크기를 조정하여 전체 상품을 처음부터 다 로드될 수 있도록 변경하여 수정했다.

DB에 향수 id, 향수이름, 향수 가격, 향수 노트, 정보와 향 정보, 향수 키워드, 향수 이미지, 향수 브랜드를 저장 후 백엔드와 연결하여 id를 통해 해당 제품의 상세 페이지로 이동한다.

ert 창이 뜨면서 장바구니 페이지로 이동한다.



장바구니와 관련된 데이터를 전역적으로 관리하고 다양한 컴포넌트 간에 쉽게 공유하기 위해 `cart-context`를 사용했다.

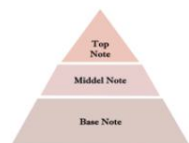


폼을 다루는 라이브러리 Formik과 입력 값 검증 라이브러리 yup를 사용하여 기능 구현했다.



React-spring 을 통해 스크롤 애니메이션을 구현했다.

항수에 대해 잘 모르는 사용자를 위해 항수의 종류와 정보에 대해 알려주는 페이지다.



함수의 참에 대해 이야기할 때, 미들, 베이스 노트 라는 용어를 사용합니다.
 노트는 시간에 따른 참의 변화를 말합니다.
 탑노트는 부리고 15분 전후로 나는 참
 미들노트는 30분~1시간 전후로 나는 참
 베이스노트는 2~3시간 전후로 나는 참
 탑노트와 베이스노트의 참은 크게 차이날 수 있으므로 충분히 시험이 필요합니다

주요 기능

(4) 로그인/회원가입

[illegible][illegible]

React Bootstrap을 이용하여 UI를 구현했다.

require을 넣어 필수로 입력할 수 있도록 구현했다.

회원이 가입 후, users 테이블에 회원정보가 저장된다.

로그인을 하면 로그인 된 회원 정보가 userInfo에 저장된다.

마이페이지로 이동하면 userInfo에 저장되어 있는 데이터를 가져와 해당 회원 정보가 뜬다.

새로고침하면 자동 로그아웃이 되는 오류를 발견했다. 메모리에 정보가 저장되지 않아 새로고침할때마다 데이터가 사라져서 로그아웃이 되는 문제였다.

해결하기 위해 메모리에 저장하고 사용자 인증 정보와 관련된 상태를 전역적으로 관리하기 위해 authcontext 모듈을 사용했다.

사용자 상관없이 무조건 주문내역에 모든 제품이 저장되는 오류를 발견했다.

해결하기 위해 사용자 아이디별로 주문한 내역을 로컬 스토리지에 저장해서 사용자별로 주문내역을 띄웠다.

문제해결 - 1. backend 파일 실행 시 함수를 찾을 수 없다는 오류가 발생할 때

해결법)

- 함수에 지정된 default 값을 지우고 다시 실행한다.

```
modelInit(Sequelize, sequelize)
^
TypeError: modelInit is not a function
    at Object.<anonymous> (C:\workspace\fumease\backe
    at Module._compile (node:internal/modules/cjs/loa
```

```
var modelInit = require("./model.js").default;
modelInit(Sequelize, sequelize)
```

문제해결 - 2. 데이터베이스 user, password가 다를 때

각자 노트북에서 DB를 다를 때, user와 password가 달라 매번 코드를 실행할 때 마다, git에 push한 이후로 코드를 수정해야 하는 번거로움이 있다.

해결법)

- <https://www.npmjs.com/package/dotenv> 에서 npm 설치 후, backend 폴더에 .env 파일을 만든다.
- env파일에 자신의 user이름과 password만 설정해준다.
- 주의사항 : git commit 시 .env 파일은 ignore 후 커밋해야 한다.

OpenAI키도 .env파일에 저장하여 키가 사라지는 현상을 방지하였음.

```
.env
backend > .env
1 DB_USER=root
2 DB_PASS=1234
```

```
backend
├── bin
│   └── www
├── node_modules
├── public
├── routes
├── views
├── .env
├── app.js
├── model.js
├── package-lock.json
└── package.json
```

```
var session = require("express-session")
const MySQLStore = require('express-mysql-session')(session);
var sessionStore = new MySQLStore({
  host: "127.0.0.1",
  port: 3306,
  user: process.env.DB_USER,
  password: process.env.DB_PASS,
  database: 'fumease'
});
```